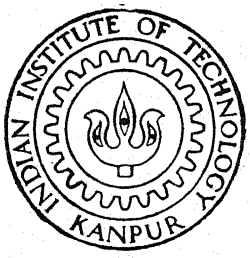
SIMULATION OF NYLON-6 POLYMERIZATION IN TUBULAR REACTORS

DEBASIS PAL



DEPARTMENT OF CHEMICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JANUARY, 1988

SIMULATION OF NYLON-6 POLYMERIZATION IN TUBULAR REACTORS

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

DEBASIS PAL

to the

DEPARTMENT OF CHEMICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JANUARY, 1988

CERTIFICATE

This is to certify that the present work 'SIMULATION OF NYLON-6 POLYMERIZATION IN TUBULAR REACTORS' has been carried out by Mr. Debasis Pal under my supervision and that this has not been submitted elsewhere for a degree.

January 1988

(Santosh K. Gupta)
Professor
Department of Chemical Engineering
Indian Institute of Technology
Kanpur

ACKNOWLEDGEMENTS

I feel deeply indebted to Professor S.K. Gupta for his untiring help and invaluable guidance and I pay my due respect to him. I am also grateful to Professor D.N. Saraf for his useful guidance during the initial period and to A. Khanna for helping me in a number of ways at various stages of my work.

I am thankful to all my friends, particularly Thomas A. Edison and Swati Karmakar, who made my stay here a pleasant and memorable one.

My thanks go J.P. Sharma and P.K. Saini for their efficient typing and to D.S. Paneser for his neat drawings.

Finally, I must make a special mention of the cooperation I received in my work from my wife, Rinka, which made this work possible.

- Debasis Pal

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A_i^O, A_i^C	Frequency factors
$A_{i,j}$	Matrix for the first derivative in orthogonal collocation
$B_{i,j}$	Matrix for the second derivative in orthogonal collocation
C_1	ϵ-caprolactam
C_2	Cyclic dimer
C_p	Specific heat, Kcal/Kg-K
$(C_p)_j$	Specific heat at the jth grid point, Kcal/Kg-K
E_i^O, E_i^C	Activation energy
h	Overall heat transfer coefficient, Kcal/m^2-K-hr
ΔH_i	Enthalpy of reaction, cal/mol
k	Thermal conductivity, Kcal/m-K-hr
k_i	Forward rate constant
k_i'	Backward rate constant
K_i	Equilibrium constant
N	Number of internal grid points (excluding the point at the wall)
P_1	Aminocaproic acid
P_n	Polymeric species of chain length n
P_n'	Symmetric Legendre polynomial of degree n
ΔQ	Overall heat of generation, Kcal/Kg-hr
r	Radius, m
r_j	Radial distance of jth grid point, m
Δr	Distance between two grid points in finite difference (equal spacing), m
R	Reactor radius, m

R_g	Universal gas constant, cal/mol-K
ΔS_i	Entropy of reaction, eu
t	Average residence time, hr
T	Temperature, K
$\bar{T}$	Average temperature at any cross-section, K
T_J	Coolant temperature in jacket, K
$U_{i,j}$	ith variable at jth grid point
v_z	Axial velocity, m/hr
v_{av}	Average axial velocity, m/hr
W	Water
W_j	Weightage vector in orthogonal collocation
x	Non-dimensional radius
x_j	Non-dimensional radius at jth grid point
z	Axial distance, m

Greek Letters:

λ_k	kth moment for the polymeric species, mol/Kg
ρ	Density, Kg/m^3
ρ_j	Density at jth grid point, Kg/m^3
μ_n	Number average chain length
$\bar{u}_n$	Average value of the number average chain length at any cross section

Subscript:

0	Initial value in the feed

Symbol:

[ ]	Concentration of a species, mol/Kg

ABSTRACT

This work presents a comprehensive computer simulation for the hydrolytic polymerization of ϵ - caprolactam without water removal in a continuous tubular reactor. This incorporates the existence of a laminar (Hagen-Poiseuille) velocity profile and radial thermal diffusion. Solution of the balance equations are carried out using two types of finite difference techniques as well as the method of orthogonal collocation coupled with GEAR's method for ODE's. The simulation results are obtained for isothermal, adiabatic and non-adiabatic operations for both plug and laminar flow velocity profiles. Some of the results were compared with available results for simpler cases. The influence of various operational variables on the reactor performance was studied. A comparison is made of the different methods used and their limitations discussed. The first method of finite difference (method I) was found to be superior to the second one, though both of them gave the same results for higher number of grid points. The orthogonal collocation technique gave good results except for the laminar flow adiabatic and non-adiabatic cases where the temperature gradients near the wall were steep. An improvement is suggested by including the finite element technique.

CHAPTER 1

INTRODUCTION

The hydrolytic polymerization of ε -caprolactam is a very important commercial process and has drawn the attention of various researchers in recent years[1-16]. In fact several excellent reviews[1-5] have appeared on this subject, emphasizing various aspects of the polymerization process.

The work on nylon-6 polymerization falls broadly into two categories. The emphasis in the early work was primarily on the determination of the polymerization mechanism and of the rate and equilibrium constants for the various reactions. These have been reviewed by Reimschuessel[1]. More recently the emphasis has shifted to the study of physical processes like heat transfer, mass transfer etc., associated with chemical reaction in large scale reactors, and to the actual modeling and optimization of industrial reactors. Tai and Tagawa[2], and Kumar and Gupta[3,4] have reviewed these aspects.

The reaction mechanism comprises of three major reactions, namely, the ring opening of caprolactam by water to form amino-caproic acid, and chain and step growth reactions. In addition, there are several important side reactions. Among these is the formation of cyclic oligomers, desamination and peroxidation of caprolactam[1]. The most important side reactions are those associated with cyclic oligomers, since their presence in the

product causes problems in its processing (e.g., in spinning and molding). The kinetic scheme considered in this work is shown in Table I. This includes the main reactions and the reactions associated with the cyclic dimer. The other cyclization reactions are omitted since their rate constants are not yet available. Moreover, it is well known[3,5] that the formation of the cyclic dimer predominates, and can be used as a first-order approximation of the total cyclics present.

The reactions are known to be catalyzed by the carboxyl end groups present in the reaction mass. The apparent rate constants are of the form

$$k_i = k_i^o + k_i^c \; [-COOH] \qquad (1)$$

with Arrhenius forms being used for k_i^o and k_i^c. All the reactions in Table I are reversible in nature and the temperature dependency of the equilibrium constants are given by standard thermodynamic relations. The rate and equilibrium constants for the reactions of Table I are given in Table II. These are based on results from a series of experiments carried out by Tai et al.,[6] using a nonlinear regression analysis.

Various kinds of reactors, e.g., batch, tubular, continuous-flow-stirred-tank reactors (CSTRs) and combinations thereof, are employed in industry to manufacture nylon-6.

Table I

Kinetic Scheme For Nylon-6 Polymerization

1. Ring Opening:

$$C_1 + W \underset{k_1' = k_1/K_1}{\overset{k_1}{\rightleftharpoons}} P_1$$

2. Polycondensation:

$$P_n + P_m \underset{k_2' = k_2/K_2}{\overset{k_2}{\rightleftharpoons}} P_{n+m} + W \; ; \; n, \; m = 1,2,3 \ldots$$

3. Polyaddition:

$$P_n + C_1 \underset{k_3'=k_3/K_3}{\overset{k_3}{\rightleftharpoons}} P_{n+1} \; ; \; n=1,2,\ldots$$

4. Ring Opening of Cyclic Dimer:

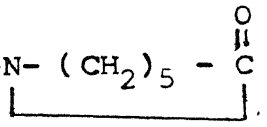
$$C_2 + W \underset{k_4' = k_4/K_4}{\overset{k_4}{\rightleftharpoons}} P_2$$

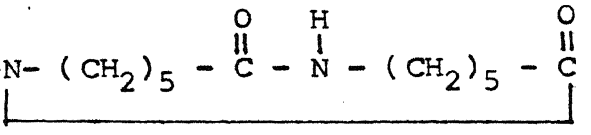
5. Polyaddition of Cyclic Dimer:

$$P_n + C_2 \underset{k_5' = k_5/K_5}{\overset{k_5}{\rightleftharpoons}} P_{n+2} \; ; \; n=1,2,\ldots$$

Contd.....

Table I (Continued)

C_1 :  (ε - caprolactam)

C_2 :  (cyclic dimer)

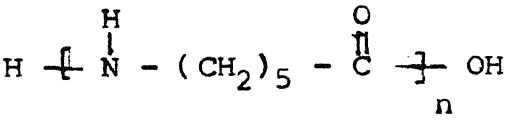
P_n :  (Polymer: Nylon-6)

Table II

Rate and Equilibrium Constants for Nylon-6 Polymerization

$$k_i = k_i^{\circ} + k_i^{c}\,[-COOH] = A_i^{\circ} \exp(-E_i^{\circ}/R_g T) + A_i^{c} \exp(-E_i^{c}/R_g T)\,[-COOH]$$

$$K_i = \exp\left\{\frac{\Delta S_i}{R_g} - \frac{\Delta H_i}{R_g T}\right\}; \quad i=1,2,\ldots,5$$

i	A_i° ($\frac{kg}{mol\text{-}hr}$)	E_i° ($\frac{cal}{mol}$)	A_i^{c} ($\frac{kg^2}{mol^2 hr}$)	E_i^{c} ($\frac{cal}{mol}$)	ΔH_i (cal/mol)	ΔS_i (eu)
1	5.9874×10^{5}	1.9880×10^{4}	4.3075×10^{7}	1.8806×10^{4}	1.9180×10^{3}	-7.8846
2	1.8942×10^{10}	2.3271×10^{4}	1.2114×10^{10}	2.0670×10^{4}	-5.9458×10^{3}	9.4374×10^{-1}
3	2.8558×10^{9}	2.2845×10^{4}	1.6377×10^{10}	2.0107×10^{4}	-4.0438×10^{3}	-6.9457
4	8.5778×10^{11}	4.2000×10^{4}	2.3307×10^{12}	3.7400×10^{4}	-9.6000×10^{3}	-1.4520×10^{1}
5	2.5701×10^{8}	2.1300×10^{4}	3.0110×10^{9}	2.0400×10^{4}	-3.1691×10^{3}	5.8265×10^{-1}

[a] Ref 2

Computer simulations based on the mathematical modeling of the polymerization processes in these reactors or their combinations offer information which is of paramount importance for quality control, process control and operational optimization of existing plants as well as in the design of new plants. Several studies have been reported on the simulation of both ideal reactors[7-9] and some common industrial reactors[10-13].

The present work focusses on the simulation of continuous tubular reactors with radial variations of temperature and concentrations accounted for. In most of the studies on tubular reactors reported till now, radial variations of temperature (and hence, of the concentrations) have not been considered. This was probably to keep the analysis simple. The only study wherein such radial profiles are indeed computed is the one by Tai et al.[7]. These workers have used the two dimensional (in the radial and axial directions) finite difference computational technique to solve their balance equations. However, they have also presented results on several other types of reactors and reactor combinations, and so their work on tubular reactors is slightly less comprehensive than what one would like to have. Moreover, our initial attempts on the optimization of such reactors has revealed the need for faster and more efficient computational techniques to integrate the modeling equations for such reactors, in order to cut down on the

total computational costs. This work was undertaken to satisfy these needs, and presents simulation results using two types of finite difference algorithms, as well as the orthogonal collocation technique. Results are generated for several conditions, e.g., for both plug and laminar flow velocity profiles in the tubular reactor and for both adiabatic and non-adiabatic operations. A detailed comparison is made of the results obtained with various computational techniques and model assumptions.

CHAPTER 2

FORMULATION

The reaction mechanism considered here is given in Table I with the rate and equilibrium constants as shown in Table II. The rates of formation of various species in the reaction mixture and those of the first two moments, λ_o and λ_1, are listed in Table III along with the rate of heat generation. In general, the k^{th} moment is defined as:

$$\lambda_k = \sum_{n=1}^{\infty} n^k [P_n] \; ; \quad k = 0,1,2..... \qquad (2)$$

The widely accepted use of moments instead of individual concentrations of polymer molecules of various chain lengths helps reduce the number of equations (for the individual species) to be solved simultaneously, and gives almost the same results[3,17-19]. At the same time, information about the first two or three moments are usually sufficient to know the nature of the molecular weight distribution in the product polymer. For example, the number average chain length (μ_n) is obtained from the first two moments,

$$\mu_n = \lambda_1 / \lambda_o \qquad (3)$$

Table IV shows the mass and heat balance equations as applied to a tubular reactor (see Fig.1) having a laminar (Hagen-Poiseuille) flow profile given by

Table III

Species Rate of Formation and Heat Generation Rate Terms

$$R_{C_1} = -k_1 [C_1][W] + k_1' [P_1] - k_3 [C_1]\lambda_o + k_3'(\lambda_o - [P_1]) \quad (a)$$

$$R_{P_1} = k_1 [C_1][W] - k_1' [P_1] - 2k_2 [P_1]\lambda_o + 2k_2' [W](\lambda_o - [P_1])$$
$$- k_3 [P_1][C_1] + k_3' [P_2] - k_5 [P_1][C_2] + k_5' [P_3] \quad (b)$$

$$R_{\lambda_o} = k_1 [C_1][W] - k_1' [P_1] - k_2 \lambda_o^2 + k_2' [W](\lambda_1 - \lambda_o)$$
$$+ k_4 [W][C_2] - k_4' [P_2] \quad (c)$$

$$R_{\lambda_1} = k_1 [C_1][W] - k_1' [P_1] + k_3 [C_1]\lambda_o - k_3'(\lambda_o - [P_1])$$
$$+ 2k_5 [C_2]\lambda_o - 2k_5' (\lambda_o - [P_1] - [P_2])$$
$$+ 2k_4 [W][C_2] - 2k_4' [P_2] \quad (d)$$

$$R_{C_2} = -k_4 [C_2][W] + k_4' [P_2] - k_5 [C_2]\lambda_o + k_5'(\lambda_o - [P_1] - [P_2]) \quad (e)$$

$$R_W = -k_1 [C_1][W] + k_1' [P_1] + k_2 \lambda_o^2 - k_2' [W](\lambda_1 - \lambda_o)$$
$$- k_4 [C_2][W] + k_4' [P_2] \quad (f)$$

ΔQ = Heat generation rate (Kcal/kg-hr)

$$= \sum_{i=1}^{5} -(\Delta H)_i R_i / 10^3 \quad (g)$$

Table III (Continued)

where,

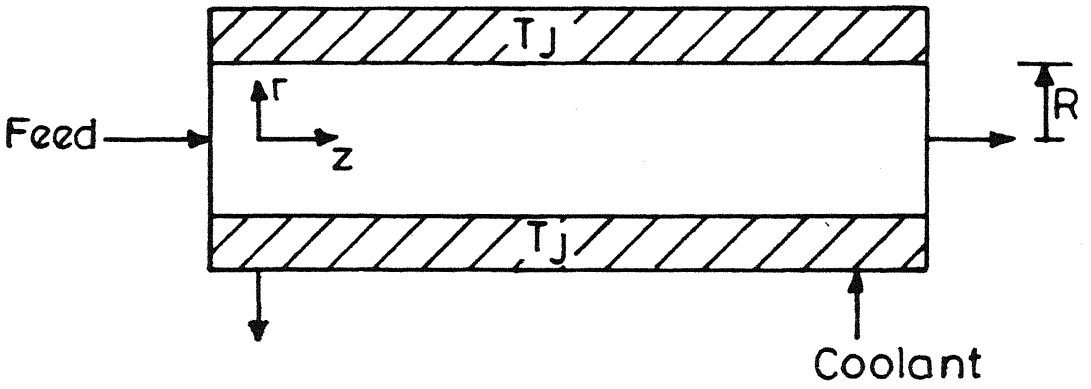
$$R_1 = k_1 [C_1][W] - k_1' [P_1] \quad (h)$$

$$R_2 = k_2 \lambda_o^2 - k_2' [W] (\lambda_1 - \lambda_o) \quad (i)$$

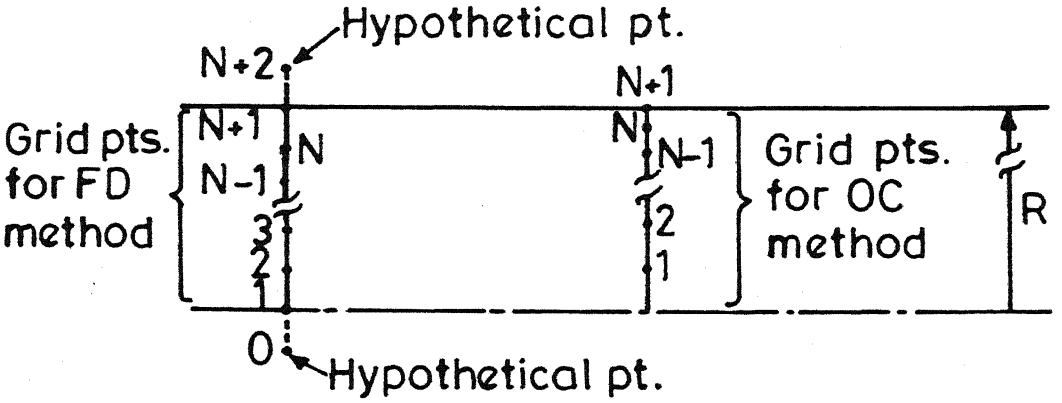
$$R_3 = k_3 [C_1] \lambda_o - k_3' (\lambda_o - [P_1]) \quad (j)$$

$$R_4 = k_4 [C_2][W] - k_4' [P_2] \quad (k)$$

$$R_5 = k_5 [C_2] \lambda_o - k_5' (\lambda_o - [P_1] - [P_2]) \quad (l)$$



(a)



(b)

Fig.1

(a) Schematic presentation of the tubular reactor with cooling jacket

(b) Grid points for finite difference (methods I & II) and orthogonal collocation methods (method IV)

$$v_z = 2v_{av} (1-r^2/R^2) \qquad (4)$$

The average residence time, t, is given by

$$t = z/v_{av} \qquad (5)$$

where z is the distance along the reactor. In writing these equations, the diffusion of all the components has been neglected and only thermal diffusion in the radial direction has been considered. While the diffusion of heavy polymer molecules may indeed be negligible, it may not be quite justified to neglect the diffusion of the lower molecular weight components, particularly, the monomer and water. This assumption is made for two reasons - firstly because precise data on the diffusion coefficients are still not available[14], and secondly as the radial diffusion in the absence of vaporization is expected to be a second order effect, and could be neglected at least for optimization studies. Tai et al.[7] have also made similar approximations. It may be pointed out here that recently some work has been reported by Hamer and Ray[20] on the detailed modeling of continuous tubular polymerization reactors incorporating radial diffusion of mass as well as property and velocity profile changes. Work along these lines for nylon-6 reactors could be carried out, but since our long range focus was on optimization studies, we did not pursue this approach.

The following correlations[7,21] have been used for the various thermophysical properties

thermal conductivity : $k = 0.21$ Kcal/m-K-hr (6a)

density : $\rho\,(Kg/m^3) = 1000\left\{1.0065 + 0.0123\,[C_1] + (T(K) - 495)(0.00035 + 0.00007\,[C_1])\right\}^{-1}$ (6b)

specific heat : $C_p\,(\frac{Kcal}{Kg\text{-}K}) = 0.6593\,\frac{[C_1]}{[C_1]_o} + \left\{1 - \frac{[C_1]}{[C_1]_o}\right\} \times \left\{0.4861 + 0.000337\,T(K)\right\}$ (6c)

The boundary condition at the wall of the reactor for the heat balance equation (eqn.h, Table IV) is obtained by considering a coolant at a constant temperature, $T_J(°K)$, circulated in a jacket enclosing the reactor (Fig. 1a), the overall heat transfer coefficient being h. The symmetry condition at r=0 is given by eqn.i, Table IV.

A radial temperature gradient will be established in the reactor at any cross-section along the length because of the presence of the velocity profile as well as the heat flux through the wall. As a consequence, the concentrations of all the species will also have a radial variation. Hence, the variables $[C_1]$, $[P_1]$, λ_0, λ_1, $[C_2]$, $[W]$, and T are all functions of r and t in general, and can be represented as $[C_1]\,(r,t)$, etc. To obtain all these profiles, the set of differential equations (eqns. a-g, Table IV) need to be

Table IV

Mass and Heat Balance Equations in a Tubular Reactor with Laminar Flow Profile

Mass Balance:

$$2(1 - r^2/R^2)\ \frac{d[C_1]}{dt} = R_{C_1} \qquad (a)$$

$$2(1 - r^2/R^2)\ \frac{d[P_1]}{dt} = R_{P_1} \qquad (b)$$

$$2(1 - r^2/R^2)\ \frac{d\lambda_o}{dt} = R_{\lambda_o} \qquad (c)$$

$$2(1 - r^2/R^2)\ \frac{d\lambda_1}{dt} = R_{\lambda_1} \qquad (d)$$

$$2(1 - r^2/R^2)\ \frac{d[C_2]}{dt} = R_{C_2} \qquad (e)$$

$$2(1 - r^2/R^2)\ \frac{d[W]}{dt} = R_W \qquad (f)$$

Heat Balance:

$$2(1 - r^2/R^2)\,\rho C_P \frac{\partial T}{\partial t} = \frac{k}{r}\frac{\partial}{\partial r}\left[r \frac{\partial T}{\partial r}\right] + \rho(\Delta Q) \qquad (g)$$

$$-k \left.\frac{\partial T}{\partial r}\right|_{r=R} = h\,(T - T_J) \qquad (h)$$

$$\left.\frac{\partial T}{\partial r}\right|_{r=0} = 0 \qquad (i)$$

$t = z/v_{av}$

integrated simultaneously with the boundary and symmetry conditions (eqns.h-i, Table IV).

A close examination of the equations of Table IV alongwith the rate expressions of Table III reveals that the equations for $[C_1]$, $[P_1]$, λ_0, λ_1, $[C_2]$, $[W]$ and T do not involve any other variable other than $[P_2]$ and $[P_3]$. The equations for $[P_2]$ and $[P_3]$ will similarly involve $[P_4]$ and $[P_5]$. To break this hierarchy of equations and to make them solvable, a simplifying assumption, called the closure approximation, is employed. This is as follows:

$$[P_3] = [P_2] = [P_1] \qquad (7)$$

This was first used by Reimschuessel[1]. Tai et al.[22] have found that the final results are insensitive to this assumption. The equations of Table IV with eqn.7 now comprise a complete set of coupled differential equations which need to be integrated numerically.

Two techniques are employed in this work to convert the partial differential equation - PDE (eqn.g, Table IV) into an ordinary differential equation - ODE. These are the finite difference (FD) and the orthogonal collocation (OC) methods. The basic principles of these methods are explained in Ref. 23.

To use the FD method, the radial distance, $0 \leqslant r \leqslant R$ is first divided into N equal parts, of length Δr each,

using (N+1) grid points numbered from 1 to N+1 (Fig. 1b). Thus the jth radial grid point at $r=r_j$ is at $r=(j-1)(\Delta r)$. The variables at r_j are renamed for the sake of convenience as follows:

$$U_{1,j}(t) \equiv [C_1](r_j,t) \; ; \quad U_{2,j}(t) \equiv [P_1](r_j,t) \; ;$$

$$U_{3,j}(t) \equiv \lambda_o \; (r_j,t) \; ; \quad U_{4,j}(t) \equiv \lambda_1 \; (r_j,t) \; ;$$

$$U_{5,j}(t) \equiv [C_2](r_j,t) \; ; \quad U_{6,j}(t) \equiv [W] \; (r_j,t) \; ;$$

$$U_{7,j}(t) \equiv T(r_j,t) \; ; \quad j = 1,2,\ldots,N+1. \qquad (8)$$

The radial derivatives in eqn.g, Table IV, the boundary condition (eqn.h, Table IV) and the symmetry condition (eqn.i, Table IV) are converted into algebraic expressions using the FD technique[23]. This has been done in two different ways. In the first method (method I), the Laplacian in the heat balance (eqn.g, Table IV) is broken up into two parts to give

$$2(1-r^2/R^2)\rho C_p \frac{\partial T}{\partial t} = \frac{k}{r}\frac{\partial T}{\partial r} + k\frac{\partial^2 T}{\partial r^2} + \rho(\Delta Q) \qquad (9)$$

The finite difference formulae[23] for the first and second derivatives (mid-point formulae) are then applied to give following ODE corresponding to the jth point:

$$2(1-r_j^2/R^2)\ \rho_j(C_p)_j \quad \frac{dU_{7,j}}{dt}$$

$$= k\ \left[\frac{1}{r_j}\ \frac{U_{7,j+1} - U_{7,j-1}}{2(\Delta r)} + \frac{U_{7,j+1} - 2U_{7,j} + U_{7,j-1}}{(\Delta r)^2}\right]$$

$$+\ \rho_j(\Delta Q)_j\ ;\quad j=2,.....,N \qquad (10)$$

At the center, i.e., at j=1 (r=0) eqn.10 presents a problem, because it involves U_7 at an undefined point, j=0, as well as involves division by zero (r_1=0). This is resolved by using L'Hôpital's rule as well as the symmetry condition in its FD form[23]:

$$\frac{U_{7,2} - U_{7,0}}{2(\Delta r)} = 0;\ \text{or,}\quad U_{7,0} = U_{7,2} \qquad (11)$$

Here, $U_{7,0}$ is the temperature at a hypothetical point, j=0 (Fig. 1b). The resulting equation corresponding to j=1 is,

$$2(1-r_1^2/R^2)\ \rho_1(C_p)_1 \quad \frac{dU_{7,1}}{dt}$$

$$= \frac{4k}{(\Delta r)^2}\ (U_{7,2} - U_{7,1}) + \rho_1(\Delta Q)_1 \qquad (12)$$

Similarly eqn. 10 for j=N+1 (r=R, at the wall) involves $U_{7,N+2}$, the temperature at another hypothetical point. This is resolved by using the FD form of the boundary condition (eqn.h, Table IV):

$$U_{7,N+2} = U_{7,N} - \frac{2(\Delta r)h}{k}\ (U_{7,N+1} - T_J) \qquad (13)$$

This is used in eqn.10 for j=N+1. The set of mass balance equations can be transformed to the FD form quite easily. Thus, 7(N+1) ordinary differential equations have been generated, which are given in Table V. The number of variables to be solved for are also 7(N+1). These ODEs are solved using a GEAR's package[23], which has a built-in step control algorithm and is particularly useful for stiff systems. The NAG Library Routine D02EBF was used for this purpose in a DEC 1090 system.

The second approach of the finite difference technique (method II) is very similar to method I except at the boundary points, j=1 and N+1. To apply the FD technique to the boundary and symmetry condition, one-sided first derivative formulae (ref. 23, p.68), which have the same degree of accuracy as the mid-point formulae used in method I, have been used. The symmetry condition at r=0 is transformed into:

$$U_{7,1} = \frac{4}{3} U_{7,2} - \frac{1}{3} U_{7,3} \qquad (14)$$

and the boundary condition at r=R gives,

$$U_{7,N+1} = \frac{hT_J - \frac{k}{2(\Delta r)} (U_{7,N-1} - 4U_{7,N})}{h + \frac{3k}{2(\Delta r)}} \qquad (15)$$

Use of the FD form of the heat balance equation at the center and boundary points is avoided by using eqn. 14 and 15.

Table V

Set of Differential Equations After Applying Method I Of Finite Difference Technique

$$2(1 - r_j^2/R^2) \quad \frac{dU_{1,j}}{dt} = (R_{U_1})_j, \quad j = 1,.....N+1 \qquad (a)$$

$$2(1 - r_j^2/R^2) \quad \frac{dU_{2,j}}{dt} = (R_{U_2})_j, \quad j = 1,.....N+1 \qquad (b)$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$2(1 - r_j^2/R^2) \quad \frac{dU_{6,j}}{dt} = (R_{U_6})_j, \quad j = 1,.....N+1 \qquad (c)$$

$$2(1 - r_1^2/R^2)\rho_1(C_p)_1 \quad \frac{dU_{7,1}}{dt} = \frac{4k}{(\Delta r)^2}(U_{7,2} - U_{7,1}) + \rho_1(\Delta Q)_1 \qquad (d)$$

$$2(1 - r_j^2/R^2)\,\rho_j(C_p)_j \quad \frac{dU_{7,j}}{dt}$$

$$= k\left[\frac{1}{r_j} \quad \frac{U_{7,j+1} - U_{7,j-1}}{2(\Delta r)} + \frac{U_{7,j+1} - 2U_{7,j} + U_{7,j-1}}{(\Delta r)^2}\right]$$

$$+ \rho_j(\Delta Q)_j \quad j = 2,.....,N+1$$

with

$$U_{7,N+2} = U_{7,N} - \frac{2(\Delta r)\,h}{k}(U_{7,N+1} - T_J) \qquad (e)$$

---

dependence of variables on t is not explicitly mentioned.

The resultant set of ODEs consists of eqn. a-c, Table V for j=1,...,N+1 and eqn.e, Table V for j=2,....,N. The total number of ODEs is 7(N+1)-2 and the number of variables is also 7(N+1)-2 . The other two variables, i.e., the temperatures at the center and at the external boundary point, can be obtained from eqn.14 and 15. The same GEAR's package is used for integrating the set of equations as well.

A compact computer package for integrating PDEs (NAG Library Routine D03PBF) was also used (method III). This package consists of a combination of the finite difference technique and the GEAR's package to solve a set of PDEs with appropriate boundary conditions. As discussed later, method I and III are essentially the same. However, method III has an additional flexibility in the sense that it can be used for any type of unequal grid spacing, upon changing an input integer, IMESH. The value used for our case was 4, which results in having more grid points near the external boundary (where the gradients are expected to be steeper) according to formula:

$$r_i = R \, \mathrm{Sin} \left(\frac{\pi(i-1)}{2N} \right) \qquad (16)$$

This method was used to generate various simulation results.

Another technique tried was the method of orthogonal collocation. The ability of this techniques to give results

of greater accuracy with less computational effort under certain conditions, is well established[23]. However, no work has been reported till now employing this efficient method for the simulation of nylon-6 tubular reactors with radial gradients. In this method (method IV), the variables are approximated as a series of orthogonal Legendre polynomials. These polynomials are symmetric in nature with respect to the independent variable which is $x = r/R$. The polynomials themselves are obtained from the orthogonality conditions e.g.,

$$\int_0^1 W(x^2)\, P_k'(x^2)\, P_m'(x^2)\, x^{a-1}\, dx = 0, \qquad k \leqslant m-1 \tag{17}$$

where, $W(x^2)$ is a weightage function which has been taken as $(1-x^2)$ in the present case. 'a' is the geometry factor and has the value of 2 for the cylindrical geometry. Once $W(x^2)$ and 'a' are fixed, all these polynomials can be obtained. The polynomial with the highest degree to be considered is $P_N'(x^2)$ where N is the number of interior collocation points (excluding the external boundary). The roots of $P_N'(x^2)$ are taken as the interior collocation points. Now the matrices $A_{i,j}$ and $B_{i,j}$ may be formed[23] from the collocation points so that, for any dependent variable y,

$$\left.\frac{dy}{dx}\right|_{\text{at ith point}} = \sum_{j=1}^{N+1} A_{i,j}\, y_j \tag{18}$$

and

$$\left.\frac{d^2y}{dx^2}\right|_{\text{at ith point}} = \sum_{j=1}^{N+1} B_{i,j}\, y_j \qquad (19)$$

Using eqn.18 in the boundary condition (eqn.h, Table IV) for our case, after nondimensionalizing the independent variable r in terms of x, we get,

$$U_{7,N+1} = \frac{-\frac{k}{R} \sum_{j=1}^{N} A_{N+1,j}\, U_{7,j} + hT_J}{h + \frac{k}{R} A_{N+1,N+1}} \qquad (20)$$

Thus, there is no need to apply the heat balance at the boundary. The symmetry condition need not be considered here as, this is automatically taken into account by the use of the symmetric polynomials. The heat balance (eqn.g, Table IV) after being written at the jth collocation point gives,

$$2(1-x_j^2)\ \rho_j (C_p)_j\ \frac{dU_{7,j}}{dt}$$

$$= \left(\frac{k}{R^2}\right) \sum_{i=1}^{N+1} B_{j,i}\, U_{7,i} + \rho_j (\Delta Q)_j,$$

$$j=1,\ldots.N \qquad (21)$$

After substituting eqn. 20 in eqn. 21 the final form of the heat balance is:

$$2(1-x_j^2)\ \rho_j (C_p)_j\ \frac{dU_{7,j}}{dt}$$

$$= \left(\frac{k}{R^2}\right) \sum_{i=1}^{N} B_{j,i}\ U_{7,i} +$$

$$\left(\frac{k}{R^2}\right) B_{j,N+1}\ \frac{\left(-\frac{k}{R} \sum_{i=1}^{N} A_{N+1,i}\ U_{7,i} + hT_J\right)}{\left(h + \frac{k}{R}\ A_{N+1,\,N+1}\right)} + \rho_j (\Delta Q)_j$$

$$j=1,\ldots,N \qquad (22)$$

The mass balance equations (eqn. a-f. Table IV) can again be applied at all the (N+1) collocation points. The resultant set of [7(N+1)-1] ODEs are listed in Table VI. From these, the [7(N+1)-1] variables (other than $U_{7,N+1}$, the temperature at the boundary) can be obtained by the use of GEAR's method with appropriate initial conditions. The temperature at the boundary ($U_{7,N+1}$) is obtained by using eqn.20.

In all the numerical techniques, the variables obtained at the radial grid points at any t are used to get the average values. The averages can be computed using the 'cup mixing rule' to give, for any concentration and moment,

$$(U_i)_{av}(t) = \frac{\int_0^R r(1-r^2/R^2) U_i \rho(r)\ dr}{\int_0^R r(1-r^2/R^2)\ \rho(r)\ dr},$$

$$\text{for } i=1,\ldots,6 \qquad (23)$$

Table VI

Set of ODEs Using Orthogonal Collocation

$$2(1 - x_j^2)\ \frac{dU_{i,j}}{dt} = (R_{U_1})_j\ ,\quad j = 1, \ldots\ldots N+1 \qquad (a)$$

$$2(1 - x_j^2)\ \frac{dU_{2,j}}{dt} = (R_{U_2})_j\ ,\quad j = 1, \ldots\ldots N+1 \qquad (b)$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$2(1 - x_j^2)\ \frac{dU_{6,j}}{dt} = (R_{U_6})_j\ ,\quad j = 1, \ldots\ldots N+1 \qquad (c)$$

$$2(1 - x_j^2)\ \rho_j (C_p)_j\ \frac{dU_{7,j}}{dt} = \left(\frac{k}{R^2}\right) \sum_{i=1}^{N} B_{j,i}\ U_{7,i} +$$

$$\left(\frac{k}{R^2}\right) B_{j,N+1}\ \frac{\left(-\frac{k}{R}\sum_{i=1}^{N} A_{N+1,i}\ U_{7,i} + h\ T_J\right)}{\left(h + \frac{k}{R}\ A_{N+1,\ N+1}\right)} + \rho_j (\Delta Q)_j\ ;$$

$$j = 1, \ldots\ldots\ldots N \qquad (d)$$

and for temperature,

$$(U_7)_{av}(t) = \frac{\int_0^R r(1-r^2/R^2)U_7 \rho(r) C_p(r) dr}{\int_0^R r(1-r^2/R^2)\rho(r)\, C_p(r)\, dr} \qquad (24)$$

In the first three methods the integrals are computed by employing a third order finite difference formula given by Gill and Miller[24]. The NAG Library Routine D01GAF is used for this purpose. In the orthogonal collocation method, the integrals are evaluated using the weightage vector generated from orthogonal polynomials[23]. The final form is as follows:

for concentration and moments,

$$(U_i)_{av} = \frac{\sum_{j=1}^{N+1} W_j(1-x_j^2)\, U_{i,j}\, \rho_j}{\sum_{j=1}^{N+1} W_j(1-x_j^2)\, \rho_j}, \qquad i=1,\ldots,6. \qquad (25)$$

and, for temperature,

$$(U_7)_{av} = \frac{\sum_{j=1}^{N+1} W_j(1-x_j^2)\, U_{7,j}\, \rho_j (C_p)_j}{\sum_{j=1}^{N+1} W_j\, (1-x_j^2)\, \rho_j\, (C_p)_j} \qquad (26)$$

In laminar flow, the residance time is infinity at the wall. To avoid numerical problems, a small slip ($v_z(r=R)/v_{av}=10^{-4}$) was assumed at the wall. The numerical results were found to

be relatively unaffected by changing the dimensionless velocity around 10^{-4}. The inlet condition (t=0) at the wall (r=R) was also modified. The equilibrium conversion values of isothermal runs at 250°C were fed in as values at r=R, t=0. This reduced the computational time significantly and avoided convergence problems. Any change in these input values was found not to affect the results, but led to longer computational times.

CHAPTER 3

RESULTS AND DISCUSSION

A general computer program was made. Results for isothermal conditions could be obtained by putting $\Delta Q=0$ in the computer program. Similarly, adiabatic operation could be simulated by putting the heat transfer coefficient, h, as 0. The program could be run using either plug flow (v_z = Constant= v_{av}) or parabolic flow (eqn.4) profiles. A typical run for t = 20 hrs and 10 FD points took 3 min 45 sec and for 10 OC points it took 2 min 36 sec on a DEC 1090 system.

The computer programs for all the four methods described previously were checked with results available in the literature.[10,25] The programs for methods I-IV were run for the simple case of an isothermal plug flow reactor with the input conditions as:

$$[C_1]_o = 8.8 \text{ moles/kg}$$

$$[W]_o = 0.16 \text{ moles/kg}$$

$$[P_1]_o = (\lambda_0)_o = (\lambda_1)_o = [C_2]_o = 0 \quad \ldots\ldots\ldots (27)$$

and for temperatures of 230°C, 240°C, 250°C, 260°C and 270°C.

The results were found to be in complete agreement with those of Ray and Gupta,[25] thus confirming the correctness of the computer programs(except the heat balance equations). The programs were also run for an adiabatic plug flow reactor with the feed coming from the top of a VK column[10]. The results were found to match

those of Gupta and Gandhi[10]. This further confirmed the correctness of our computer programs (including the heat balance equations).

Having developed confidence in the computer packages prepared, detailed temperature and concentration profiles for several conditions were generated. The feed conditions were the same as in eqn.27. Table VII gives the various flow and heat transfer conditions used. Figs.2 to 17 present profiles for monomer conversion, number average chain length(μ_n), temperature and cyclic dimer concentration at various values of t ($\equiv z/v_{av}$). For heat transfer coefficient, h, a typical value[11] of 5.0 kcal/m^2-K-hr was used. The existence of distinct concentration and temperature profiles are quite evident from these figures.

Figs. 2 to 5 show the profiles at t = 5hr for various flow and heat transfer conditions (Table VII). The laminar flow case (curves 4,5 & 6) is the most interesting. Since the velocity near the center is higher, the residence time of a fluid element near the center (at the same t) is smaller when compared to that of a fluid element near the wall. Thus the reaction is close to equilibrium at the wall. A maximum in the temperature slightly away from the wall is observed (curves 5 & 6, Fig-2) at t=5hr. This represents a combination of higher heat geneartion (due to higher near equilibrium conversion) near the wall as well as heat

Table VII

Various Flow And Heat Transfer Conditions Used In Simulation Runs

Run No.	Type of Flow	Heat Transfer Condition	Heat Transfer Coefficient, h($Kcal/m^2-K-hr$)	Feed Temp., T_o ($^\circ C$)	Coolant Temp., T_J ($^\circ C$)
1	Plug	Isothermal	-	250	-
2	Plug	Adiabatic	0	250	-
3	Plug	Non-Adiabatic	5.0	250	260
4	Laminar	Isothermal	-	250	-
5	Laminar	Adiabatic	0	250	-
6	Laminar	Non-adiabatic	5.0	250	260

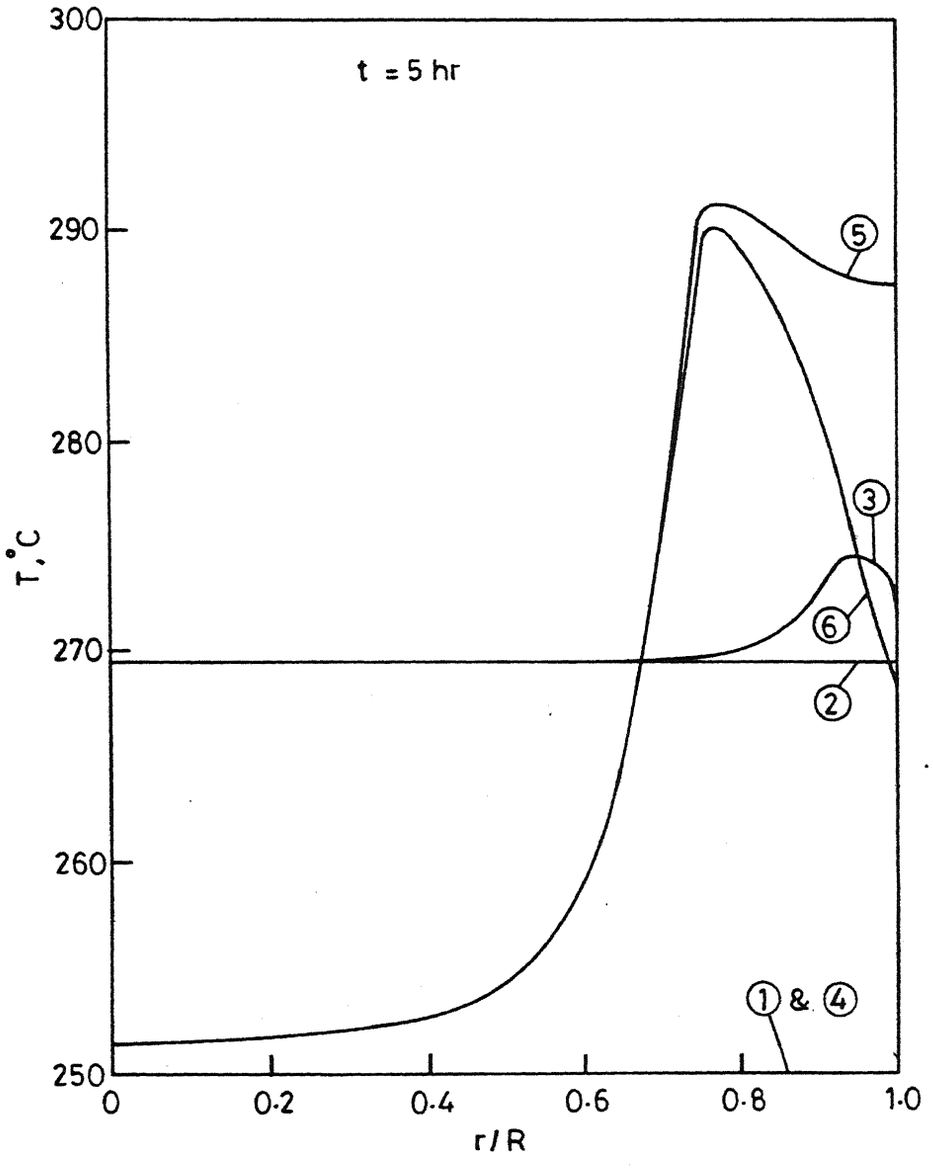


Fig. 2 .

Radial profile for temperature at t=5 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

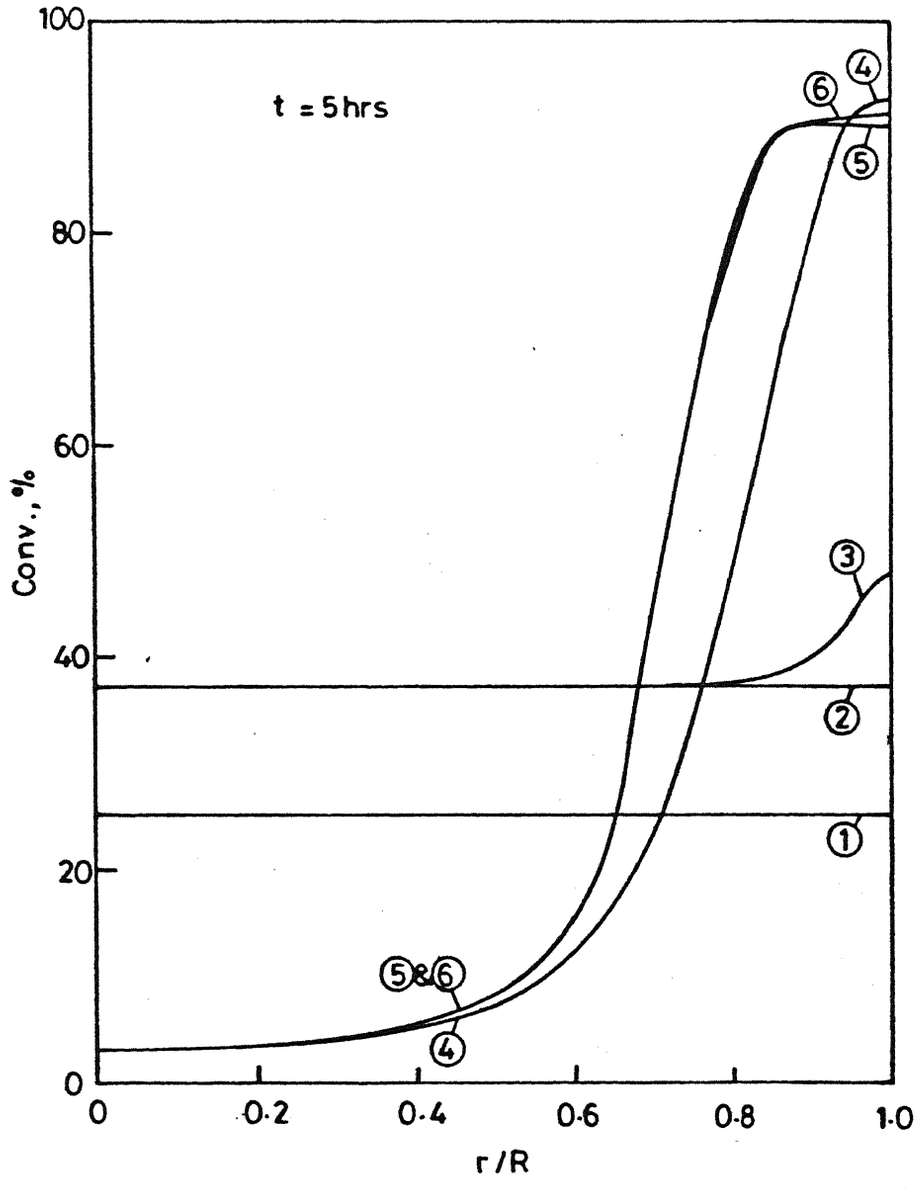


Fig. 3.

Radial profile for conversion at t=5 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

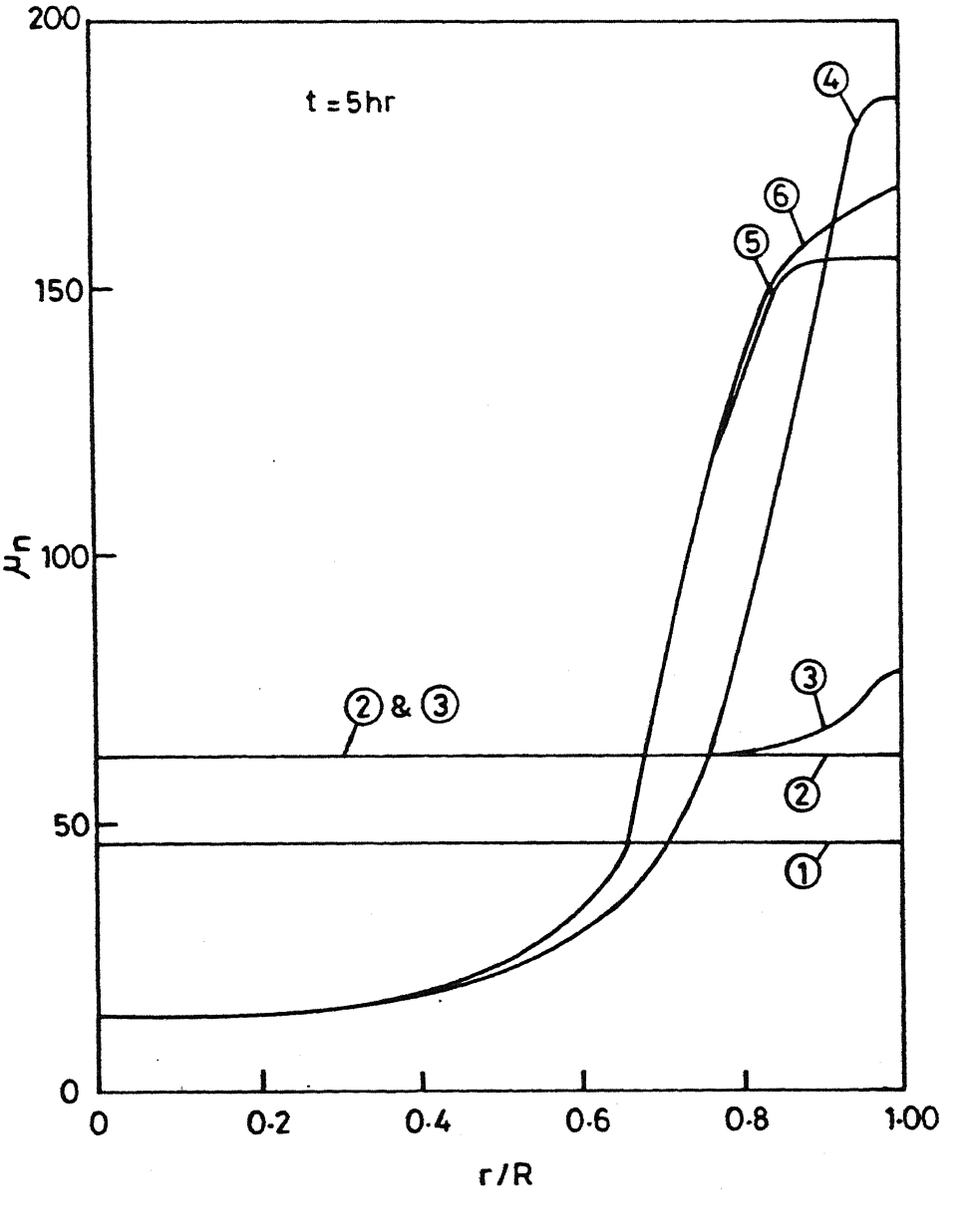


Fig. 4 .

Radial profile for μ_n at t=5 hr for various flow and heat transfer conditions (given in Table VII). Feed is as per eqn. 27.

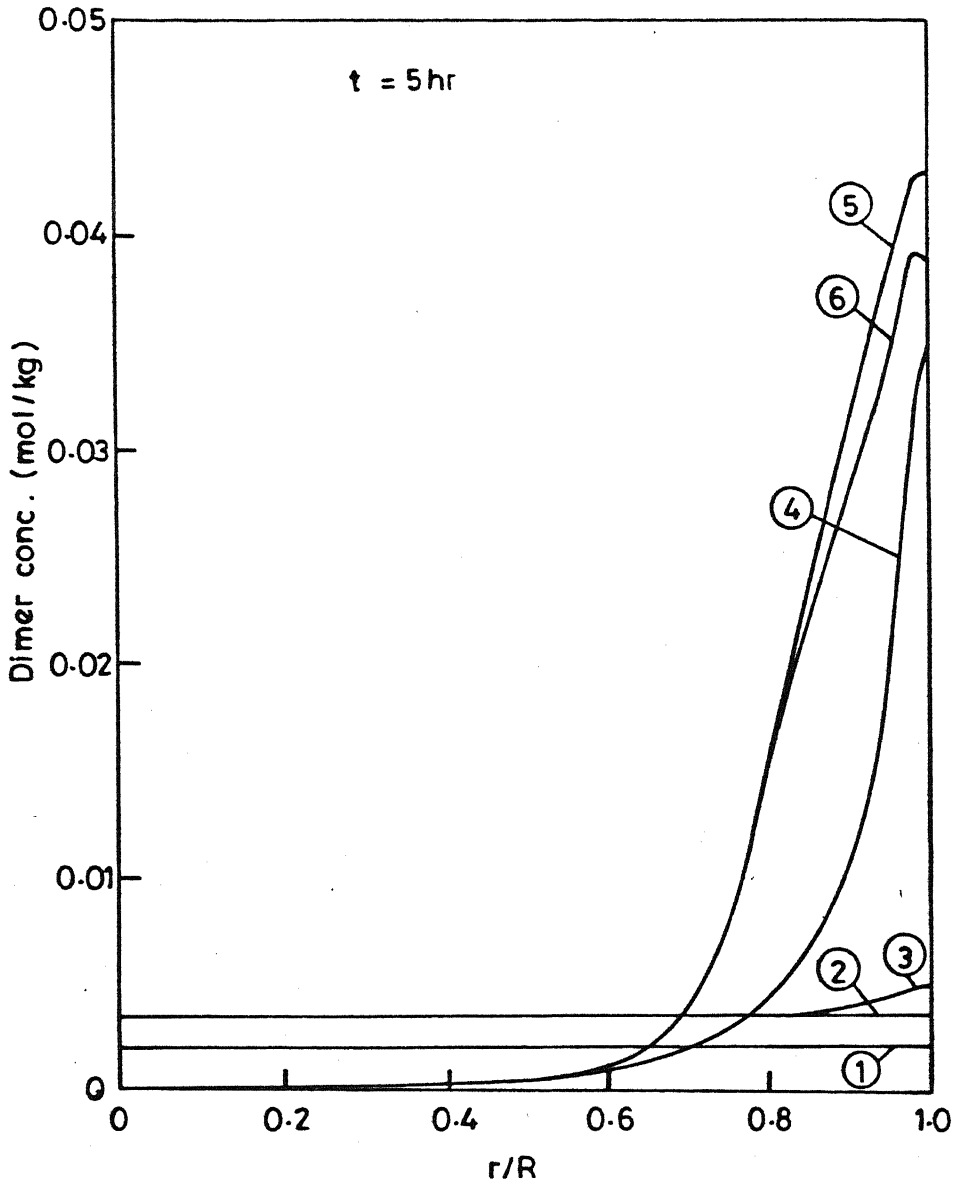


Fig.5

Radial profile for $[C_2]$ at t=5 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

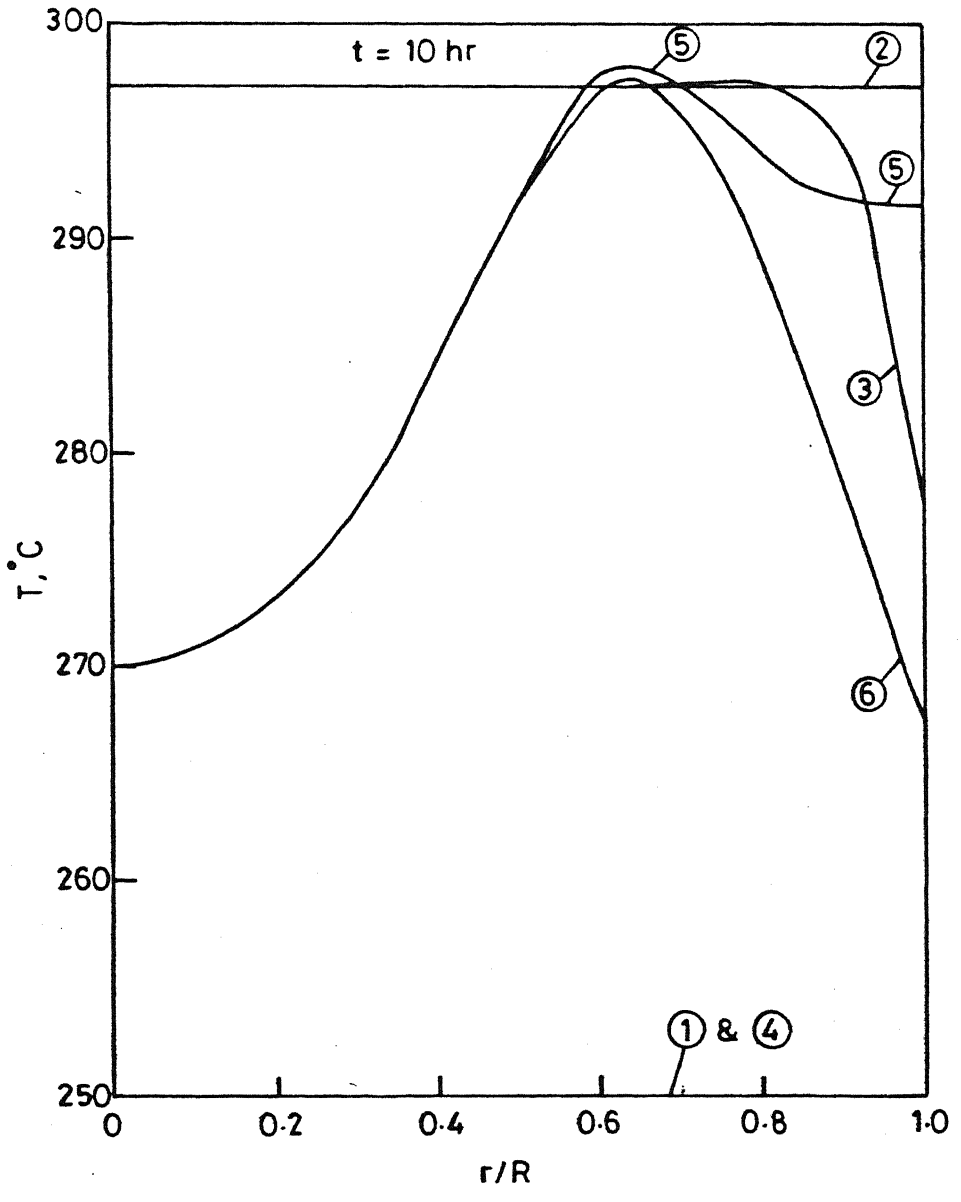


Fig. 6 .

Radial profile for temperature at t=10 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

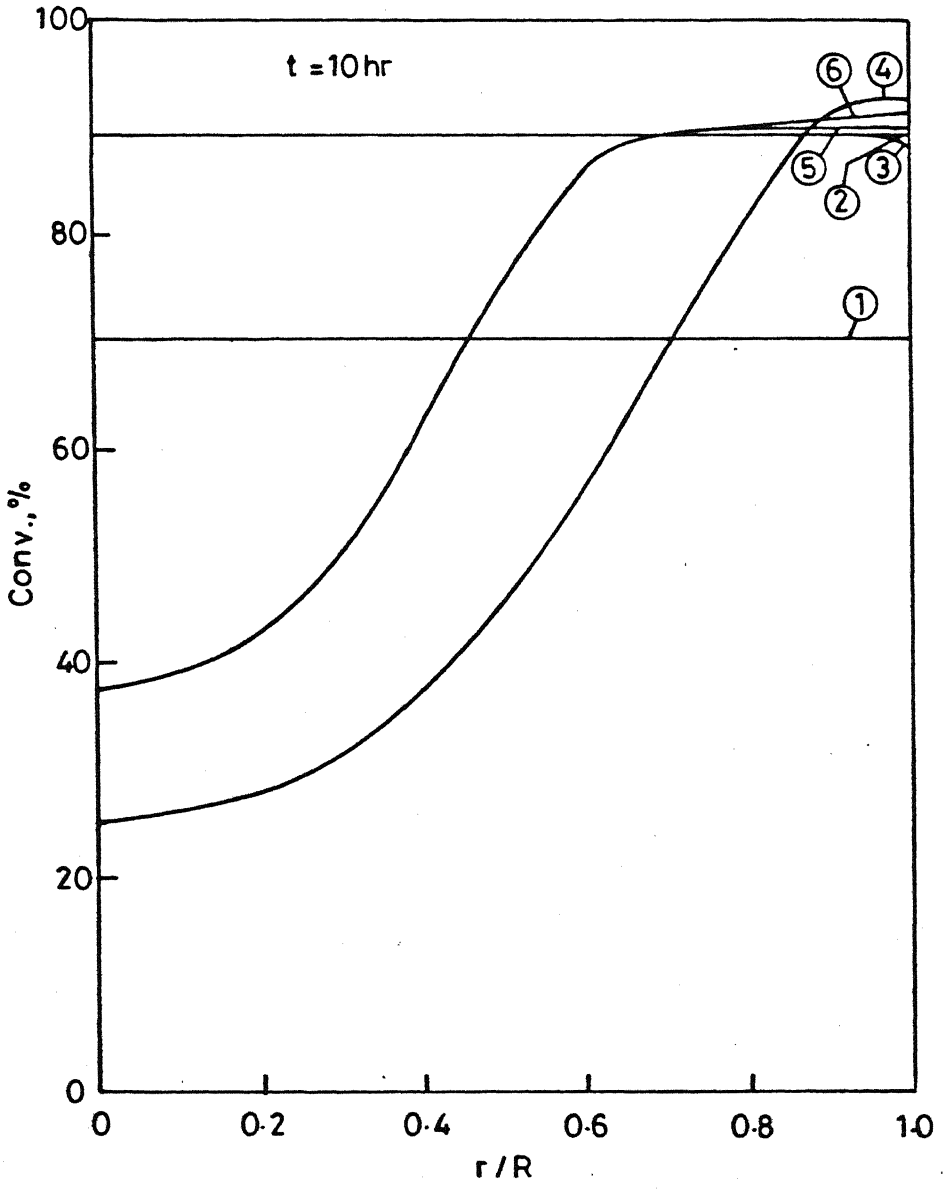


Fig. 7.

Radial profile for conversion at t=10 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

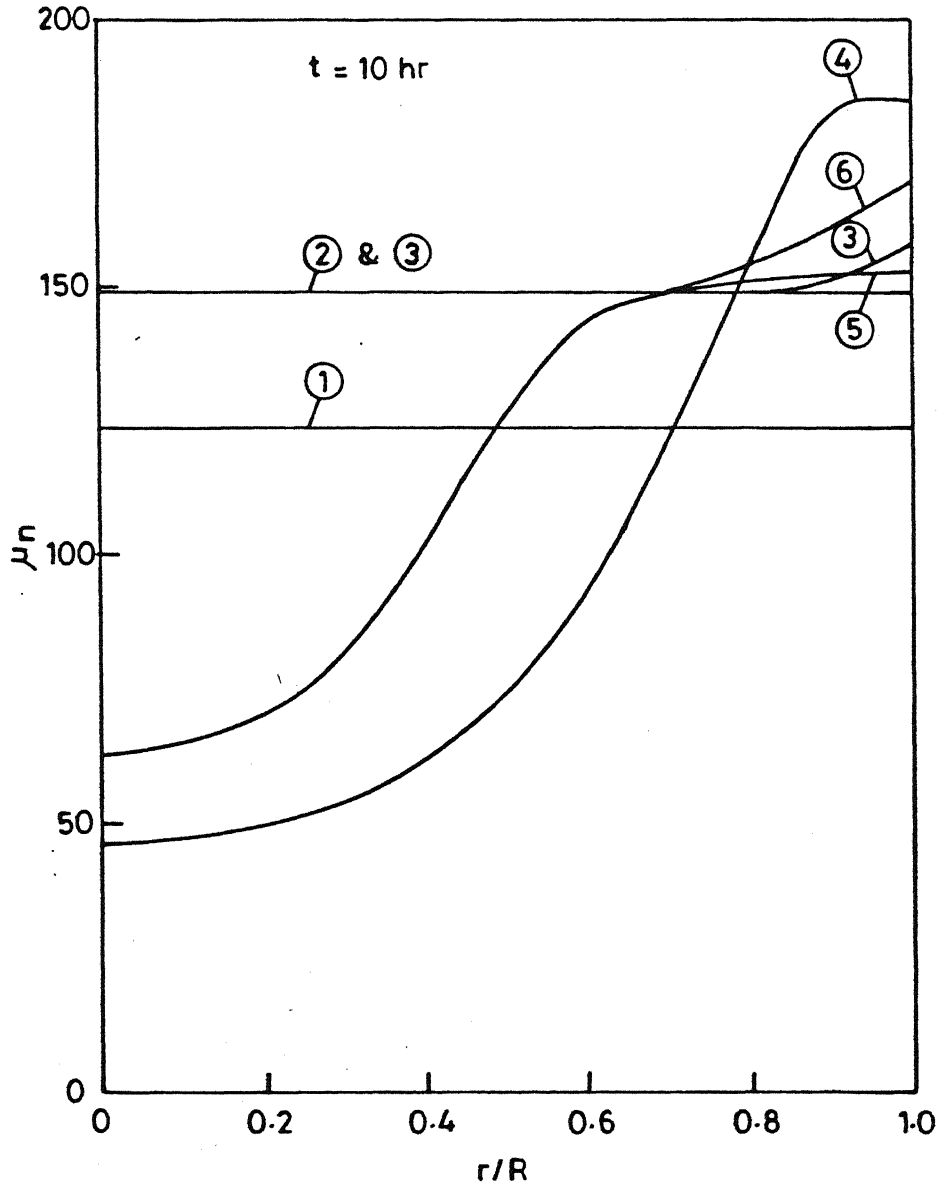


Fig.8.

Radial profile for μ_n at t=10 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

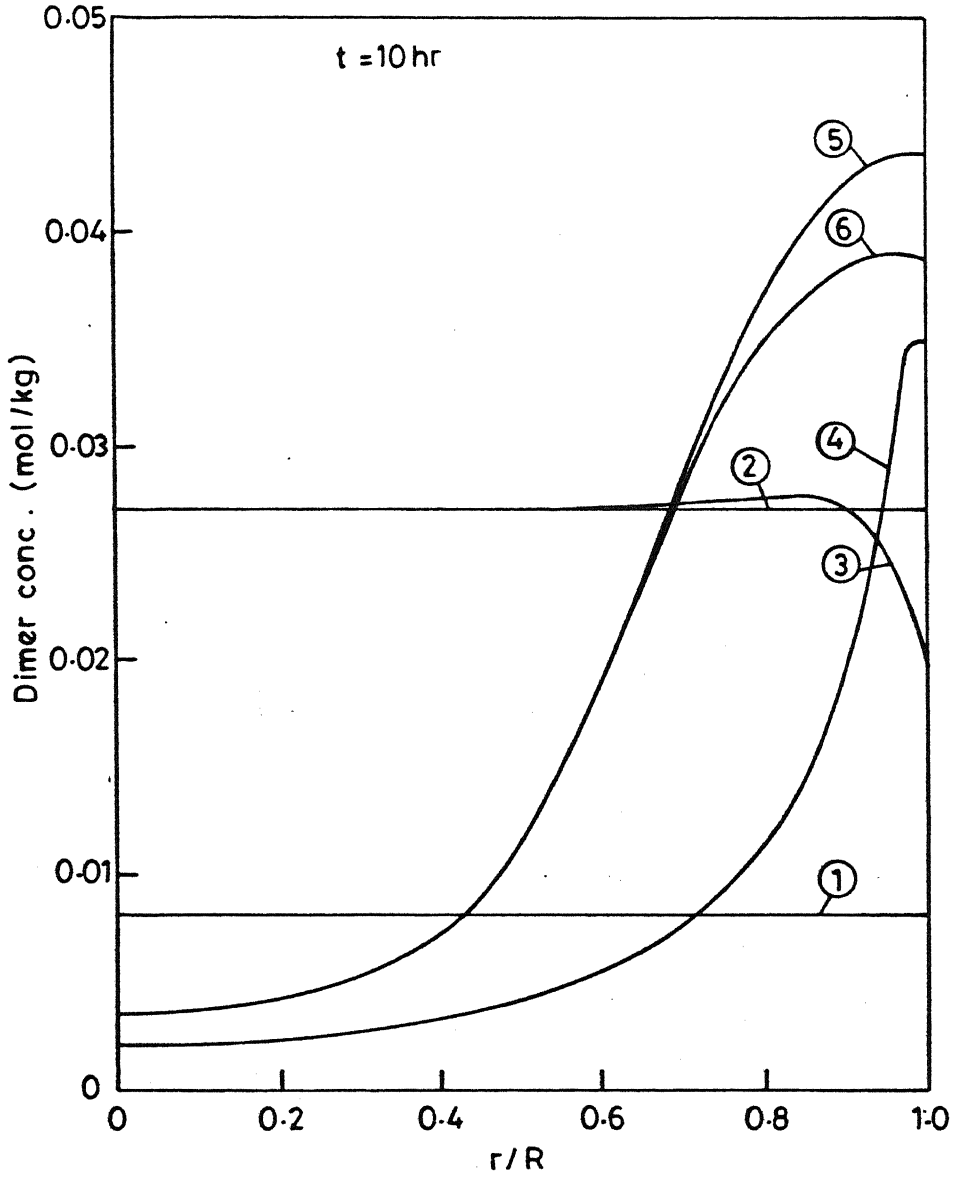


Fig. 9.

Radial profile for $[C_2]$ at t=10 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

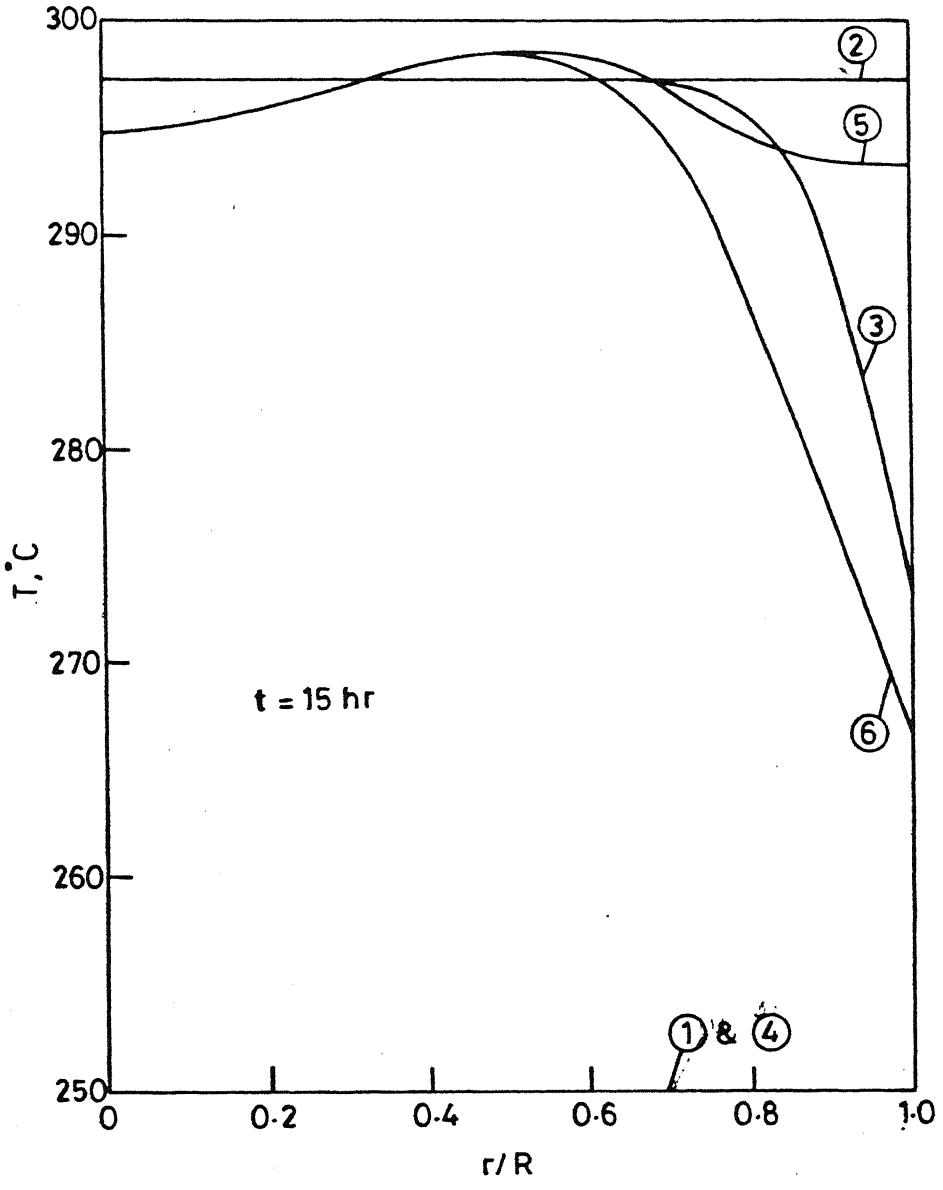


Fig. 10.

Radial profile for temperature at t=15 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

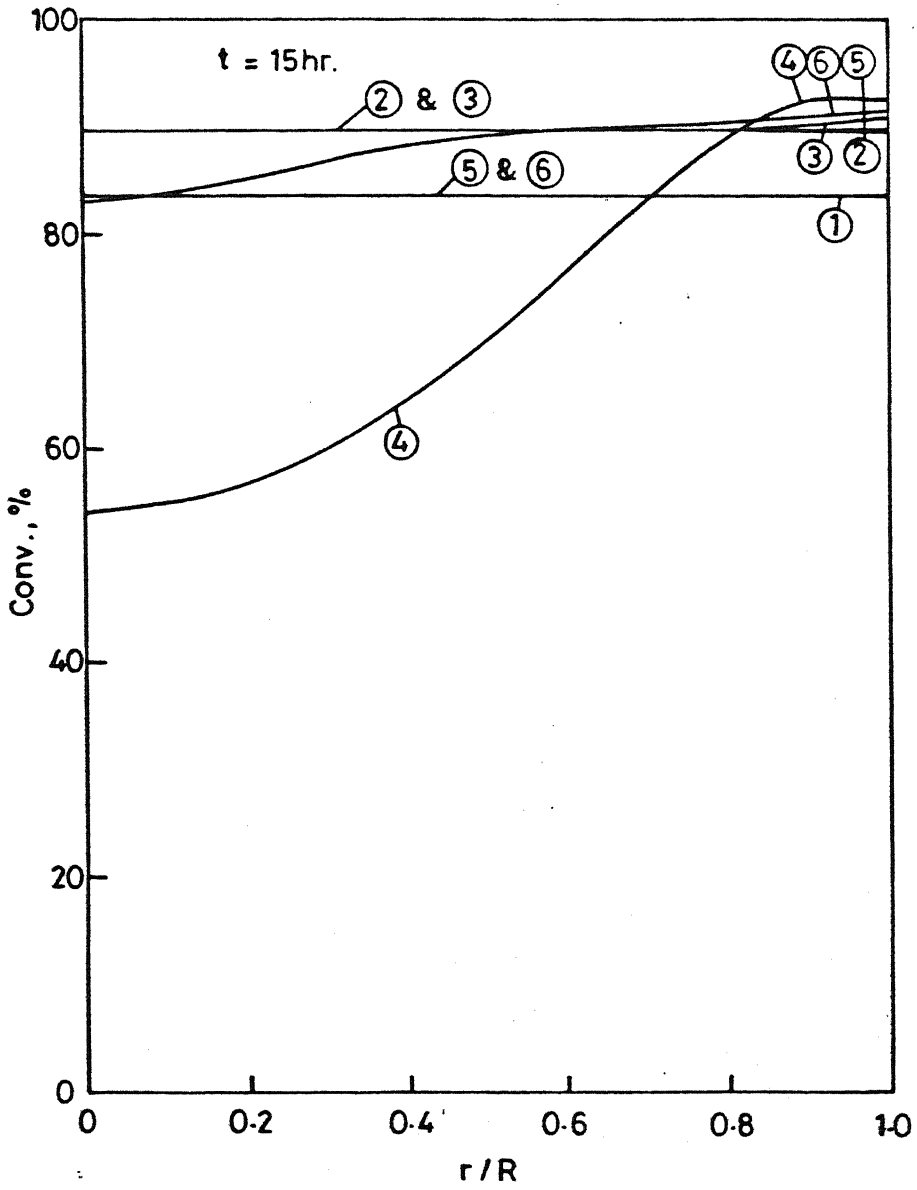


Fig.11.

Radial profile for conversion at t=15 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

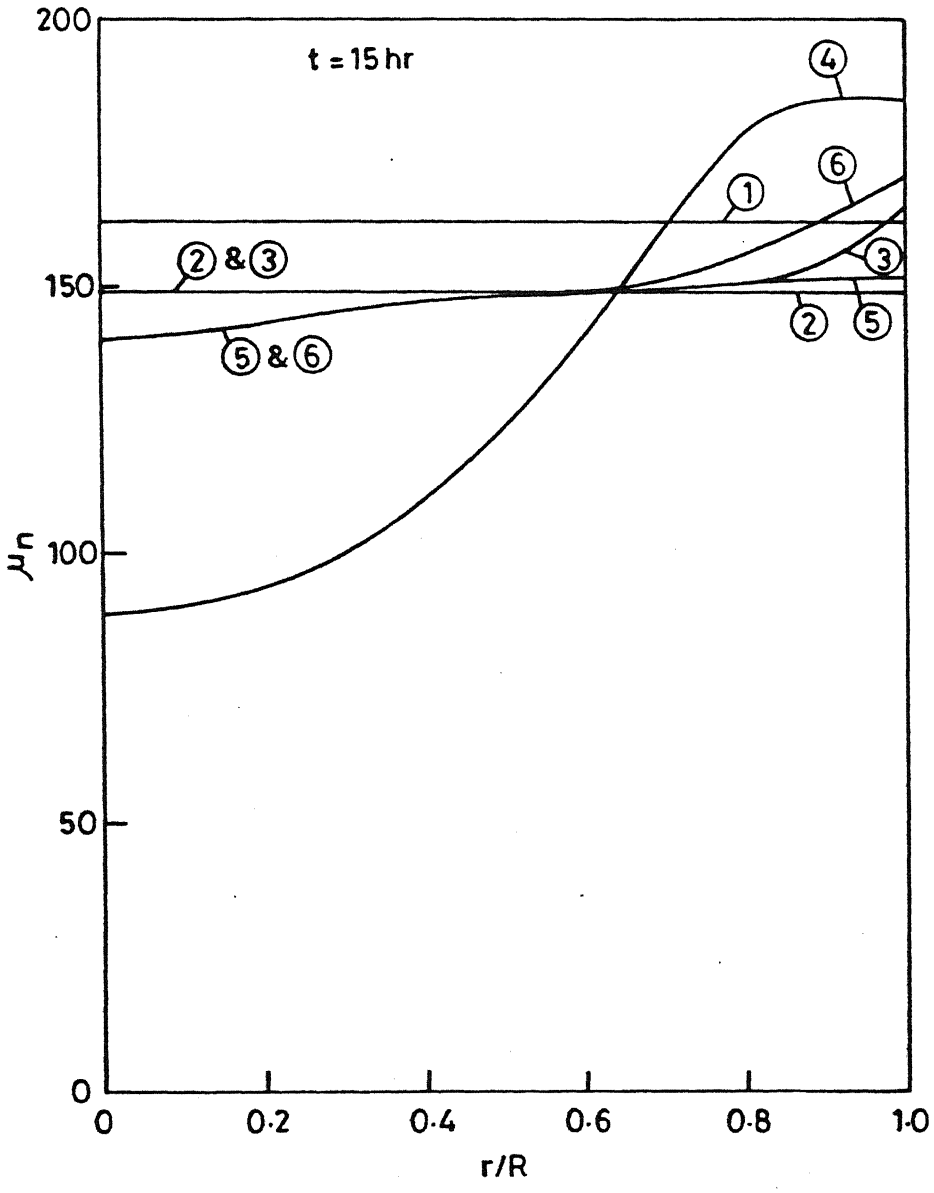


Fig.12.

Radial profile for μ_n at t=15 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

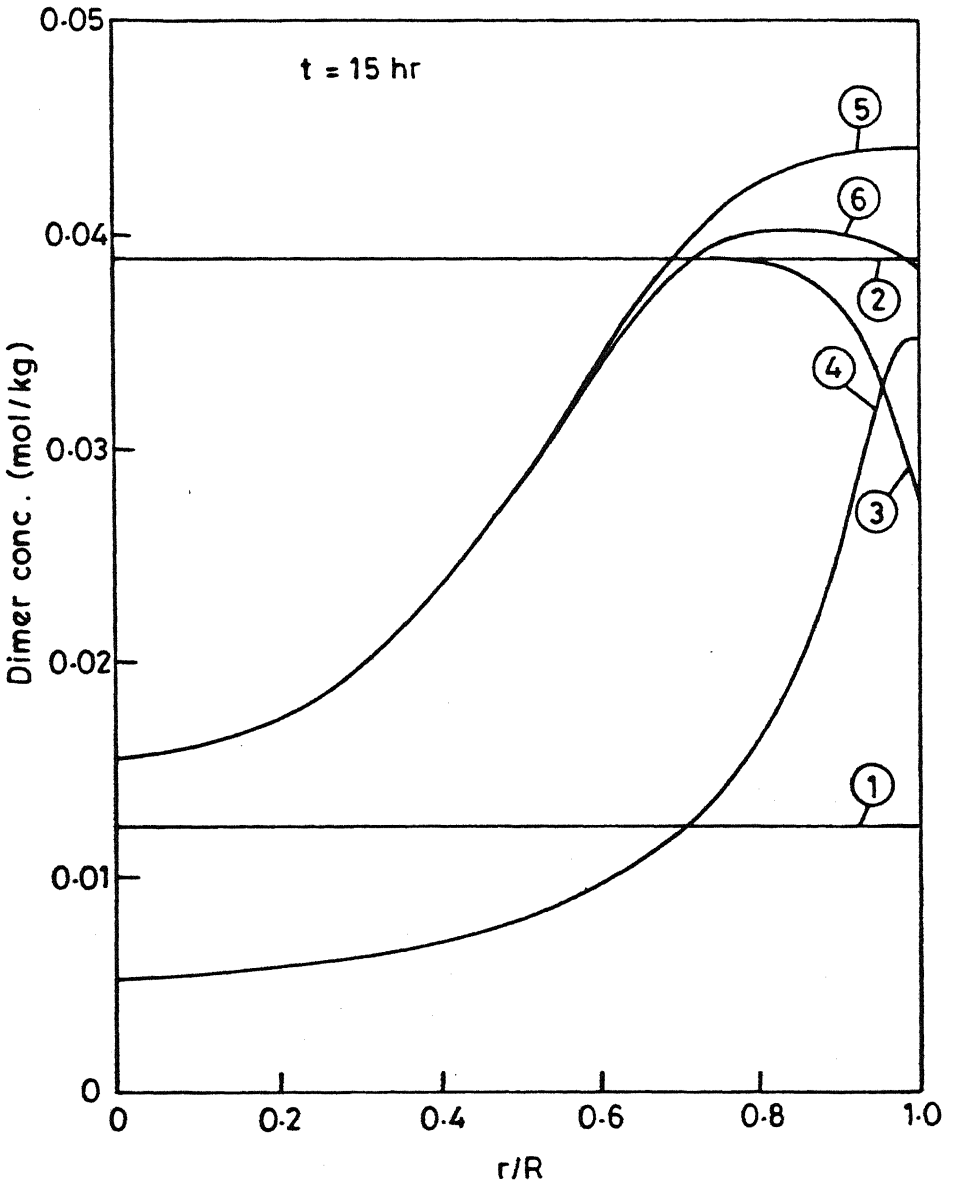


Fig. 13.

Radial profile for $[C_2]$ at t=15 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

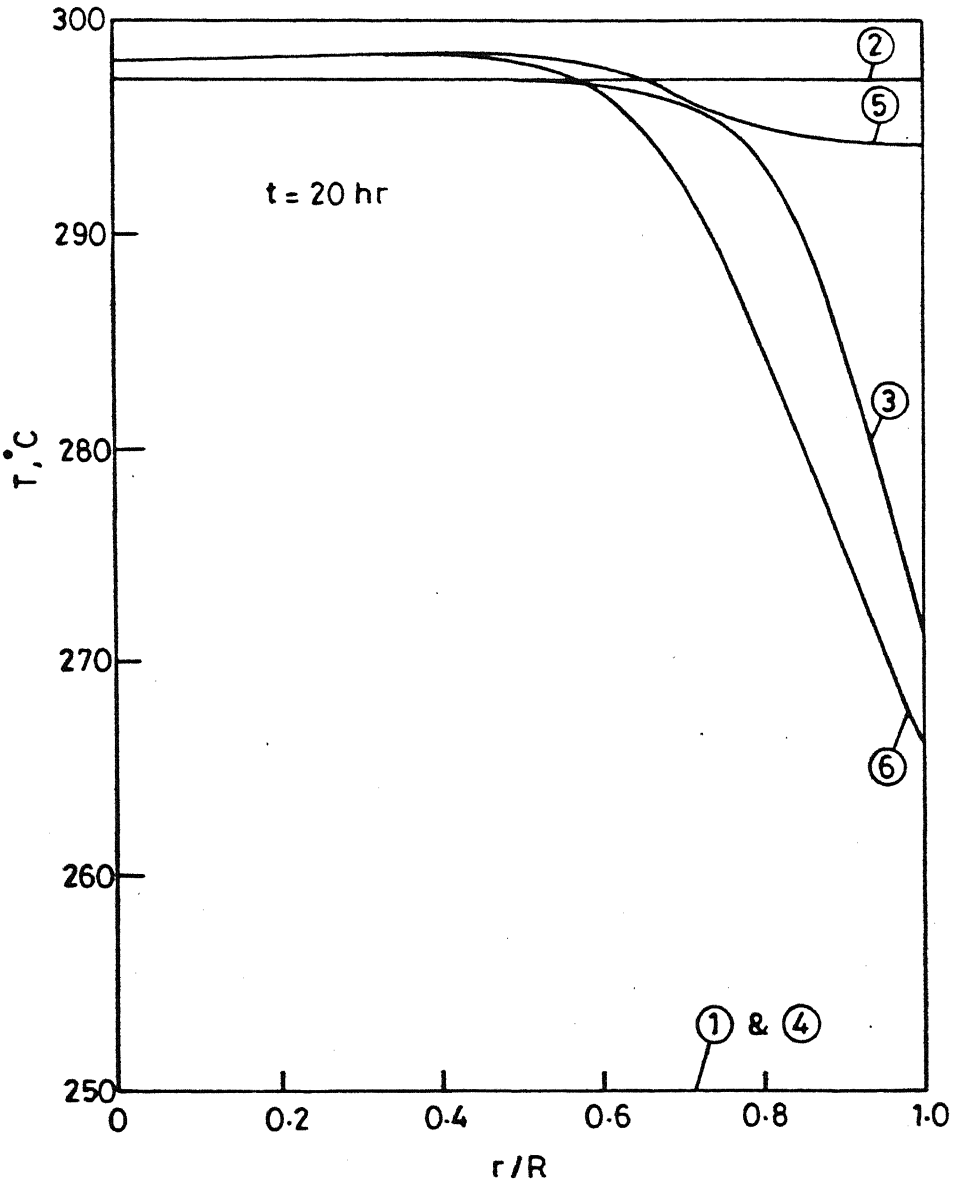


Fig.14.

Radial profile for temperature at t=20 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

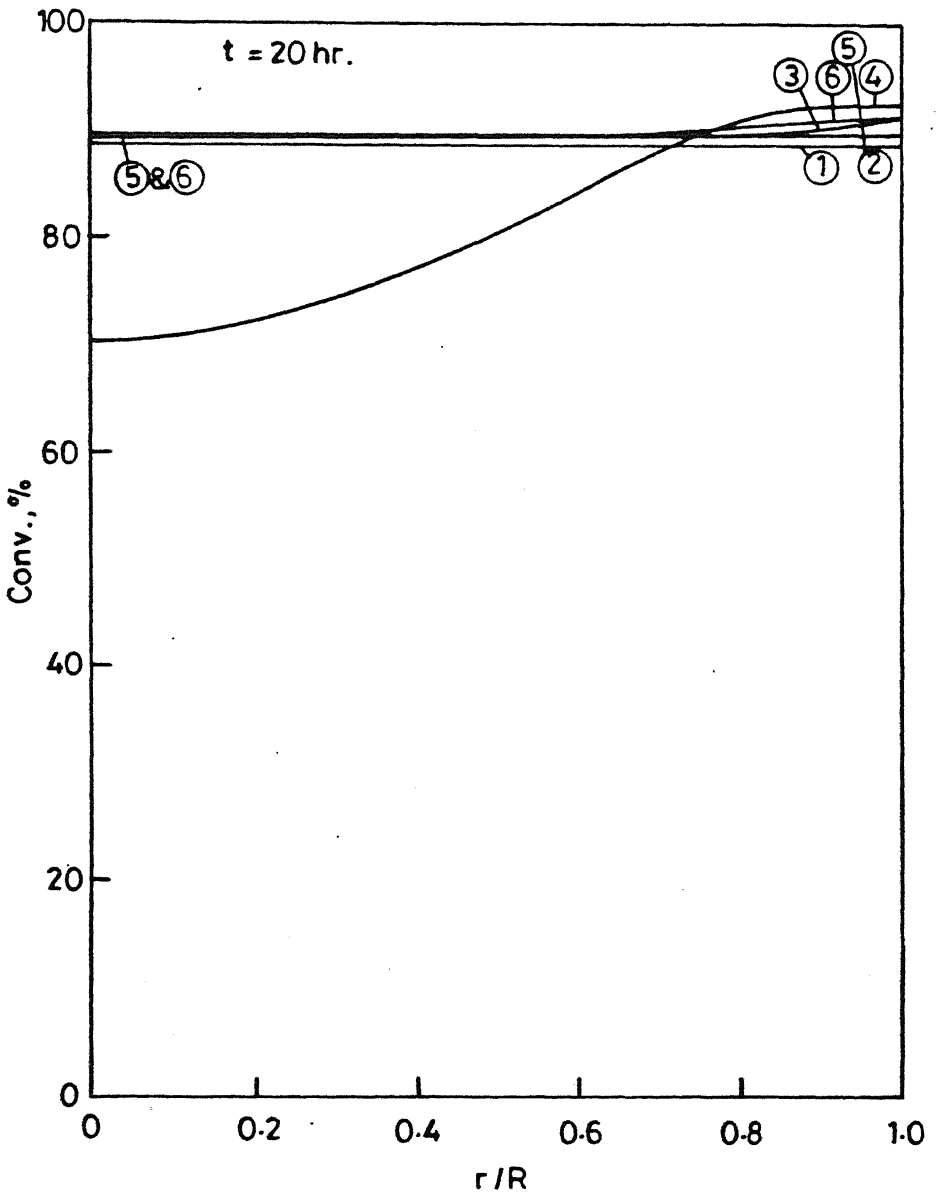


Fig.15.

Radial profile for conversion at t=20 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

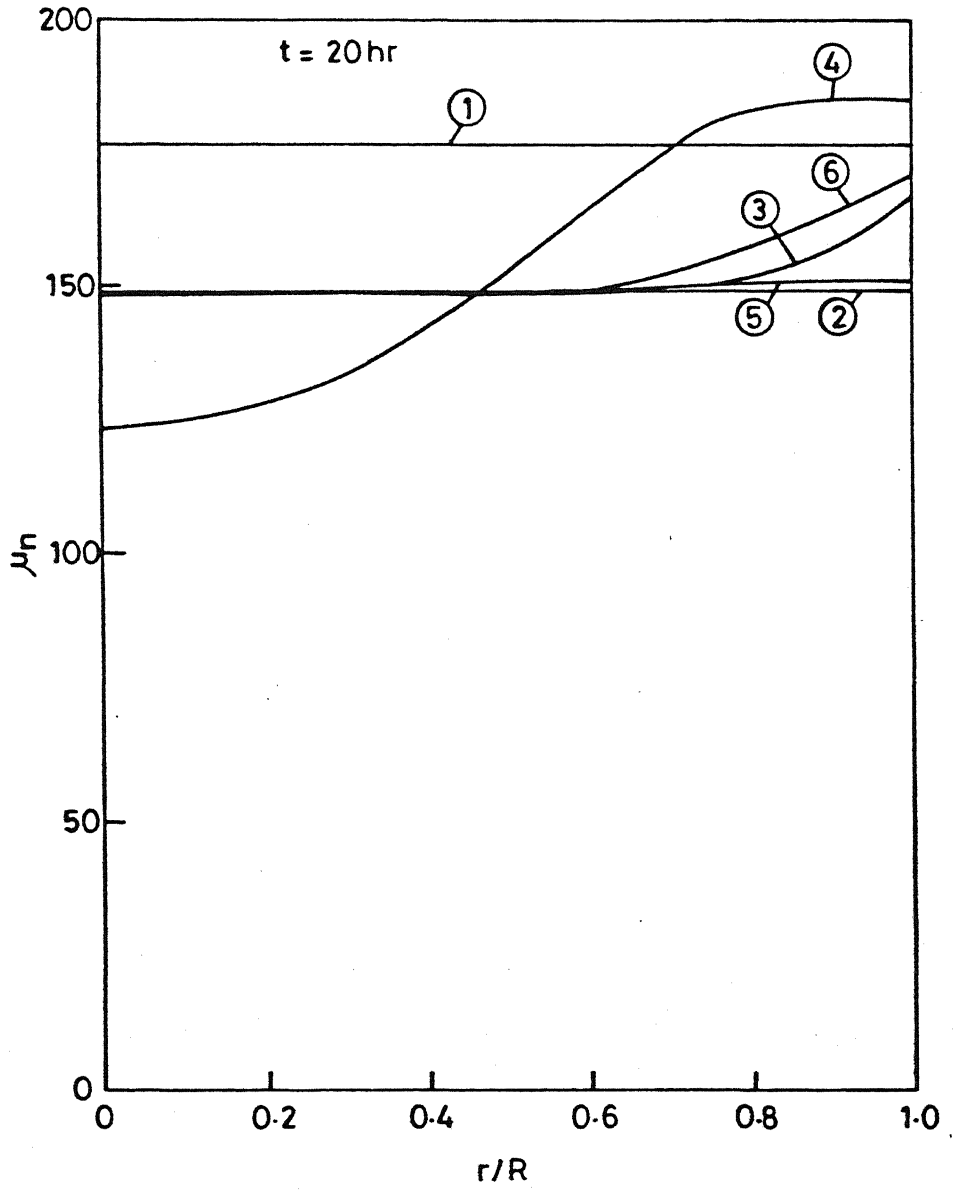


Fig.16.

Radial profile for μ_n at t=20 hr for various flow and heat transfer conditions (given in Table VII). Feed condition is as per eqn. 27.

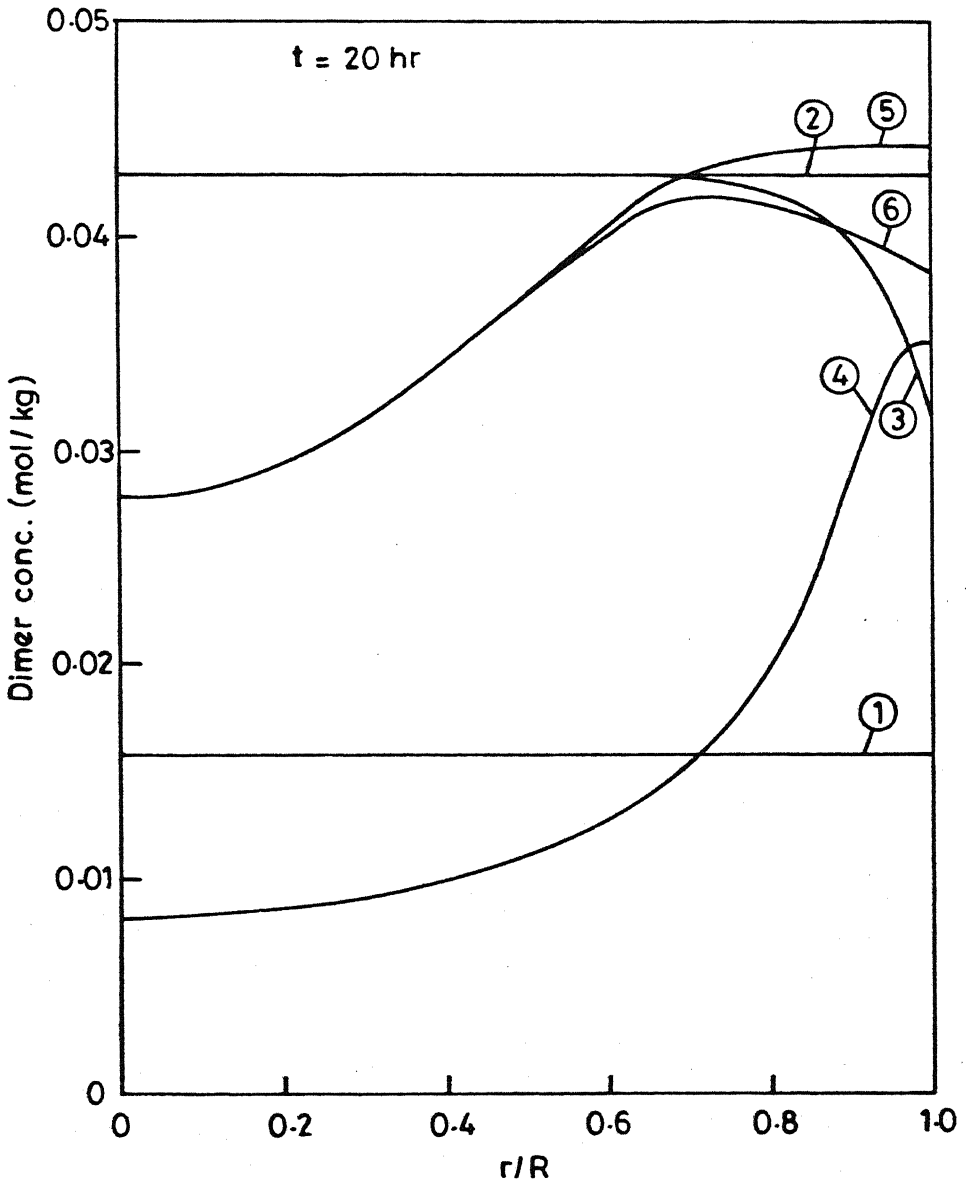


Fig. 17.

Radial profile for $[C_2]$ at t=20 hr for various flow and heat transfer conditions (given in table VII). Feed condition is as per eqn. 27.

transfer effects. In the laminar adiabatic case (curve 5, Fig.2), a temperature maximum first appears (at t near zero) at the wall because of almost instantaneous attainment of equilibrium conversions there. The higher temperature leads to thermal diffusion away from the wall as t increases (since h=0), contributing to a decrease in the temperature at the wall. In addition, as t increases layers adjacent to the wall reach near-equilibrium conversion (curves 5 & 6, Figs.3 to 5), generating more heat there. This explains the temperature maximum in curve 5, Fig.2. The peak shifts further inwards as t increases (fig.6). Because of cooling from the jacket, the temperatures at the wall are lower for the laminar non-adiabatic case (curve 6, Fig.2). The slightly higher near-equilibrium conversions for curve 6, Fig.2 (compared to curve 5), is because of the lower temperatures present and the overall exothermic nature of the reactions. Since the equilibrium values of μ_n are more sensitive to temperature, the difference between curves 5 and 6 (Fig.4) are greater. In plug flow, the residence times of fluid element at the wall are not infinite, and so, near-equilibrium conditions do not prevail at the wall for this ideal reactor. In the adiabatic case, the temperature (and other) profiles are flat as expected, since there is no driving force for creating radial differences. For the plug flow non-adiabatic case (curve 3, Fig.2) a temperature maximum near the wall develops again. Near t=0, a maximum appears at the wall ($T_J > T_o$). This beds to higher conversions at the wall, which simultaneously leads to higher heat generation by reaction at that point. Very soon the temperature

at the wall increases beyond T_J. Heat transfer takes place both to the jacket as well as towards the center. The conversion is the highest at the wall for case 3 (Fig.3) since it represents a cumulative effect from t=0 to 5. Figs. 6 to 17 show conditions at t=10, 15 and 20 hr. The trends at t=5 hr continue, but some flattening of temperature profiles in observed, because of thermal diffusion effects away from the peak, both towards the wall as well as towards the center. The conversion, μ_n and cyclic dimer profiles show higher levels near the center as t increases, as expected. Figs. 15 and 17, at t=20hr, show that even though the monomer conversion has almost attained equilibrium values throughout the cross-section, lower cyclic dimer concentrations near the center are observed. This is because equilibrium for the cyclic dimer reactions has not yet been attained.

Figs. 18 and 19 show the profiles of average temperature, $\bar{T}$, and average chain length, $\bar{\mu}_n$, along the reactor length for various flow and heat transfer conditions. It is interesting to note that the distinct "S' - shape for the plug flow case (curves 2 & 3, Fig.18) is relatively less prominent in the laminar flow case (curves 4 & 5, Fig.18). Average temperatures for the non-adiabatic ease (curves 3 & 6, Fig.18) are in general lower than that for the adiabatic case (curves 2 & 3, Fig.18) because of the cooling effects of the jacket fluid, except for the initial part of the plug flow case (curves 2 & 3) where the jacket fluid has a heating effect as discussed earlier. For $\bar{\mu}_n$, again, the

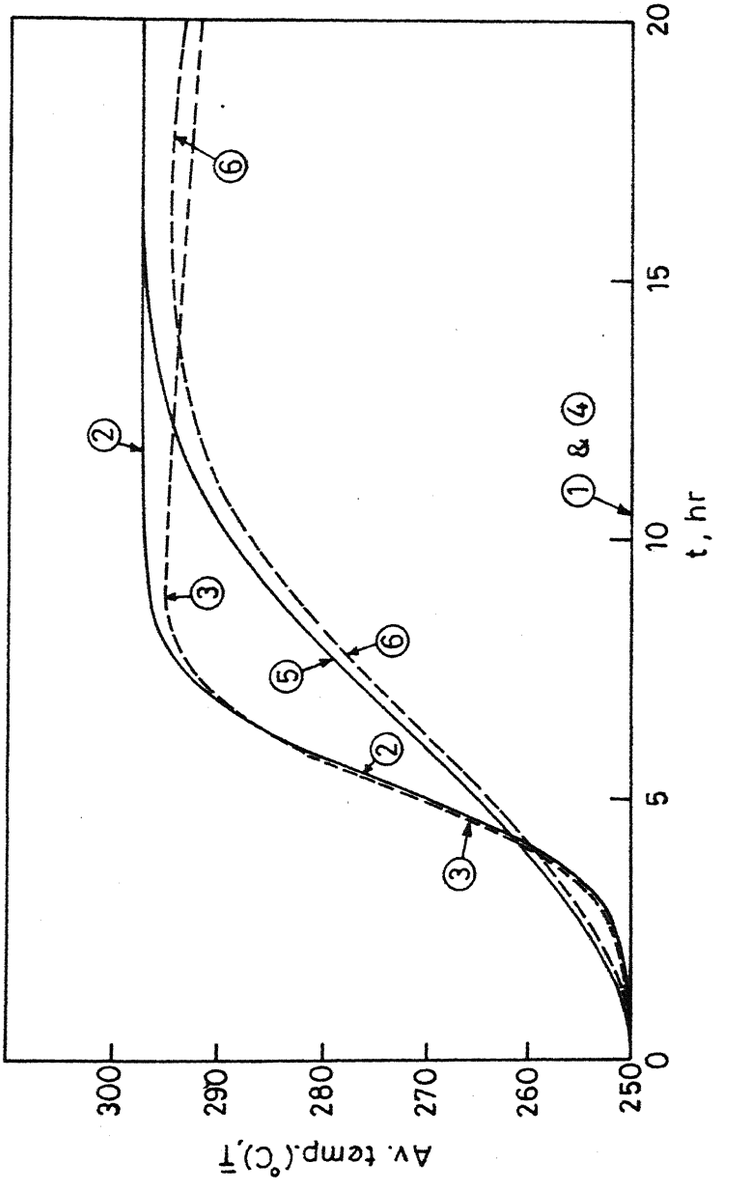


Fig.18.

Axial variation of $\bar{T}$ for various flow and heat transfer conditions (given in Table VII) with feed condition as per eqn. 27.

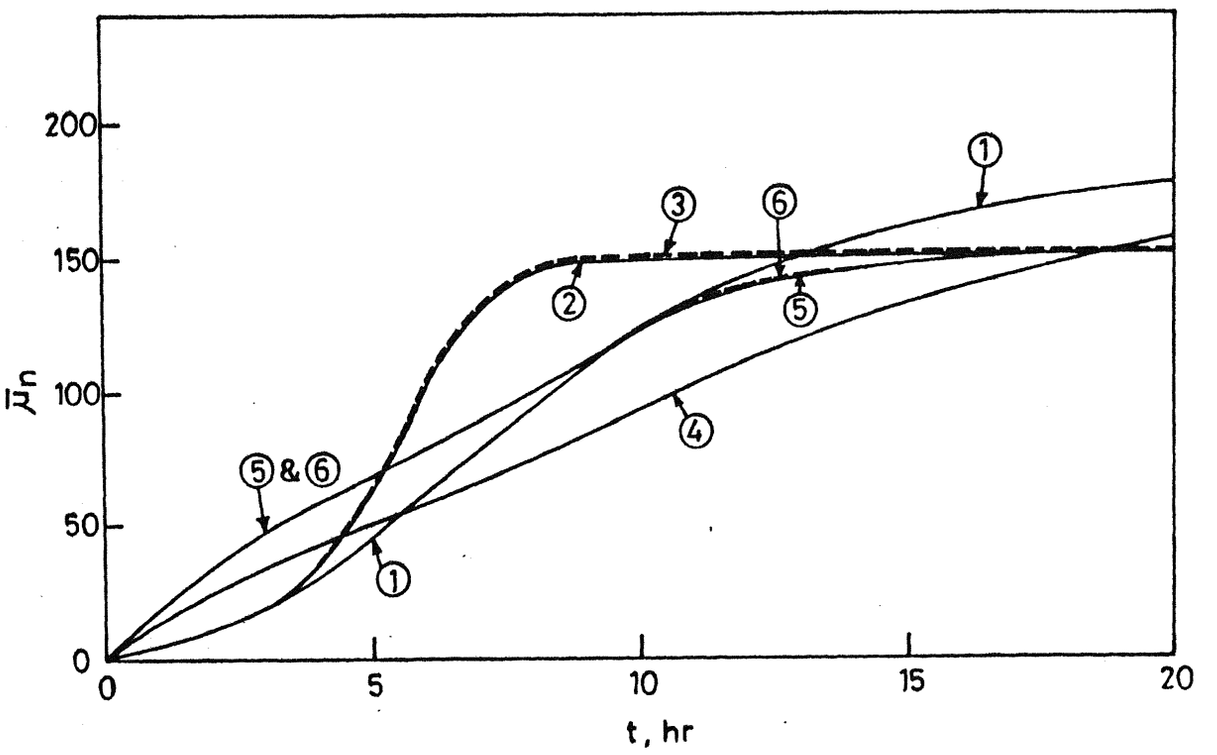


Fig.19.

Axial variation of $\bar{\mu}_n$ for various flow and heat transfer conditions (given in Table VII) with feed condition as per eqn. 27.

'S' - shape in case of plug flow (curves 1,2 & 3, Fig.19) is less prominent in laminar flow (curves 4,5 & 6 Fig.19). In general, isothermal operation (curves 1 & 4, Fig.19) gives lower $\bar{\mu}_n$ in the initial part of the reactor because of comparatively lower temperatures in this case. But as t increases and equilibrium conversions are attained, $\bar{\mu}_n$ for the isothermal case registers higher values.

Table VIII shows the effects of the various parameters, e.g., radius of the reactor (R), coolant temperature (T_J), feed temperature (T_o) and inlet water concentration ($[W]_o$), on the average values of temperature, conversion, μ_n and cyclic dimer concentration. Decreasing the value of R to 0.4 (Run-2) from the reference conditions (Run-1) helps increase radial heat transfer. Thus, the initial heating effects of the jacket fluid increases $\bar{T}$ at 5 hrs over that for the reference run. As t increases, the temperature is comparatively lower due to better cooling. However, average conversion and $\bar{\mu}_n$ are always lower. Increasing the value of R (Run-3, Table VIII) has the opposite effects. A lower coolant temperature (Run-4, Table VIII) causes lower temperature in the reactor. This reduces the average conversion and $\bar{\mu}_n$ initially. But, as equilibrium is attained at higher values of t, the average conversion and $\bar{\mu}_n$, are higher as an effect of lower temperatures. A lower feed temperature (Run-5, Table VIII) causes lower $\bar{T}$. As before, average conversion and $\bar{\mu}_n$ are initially lower, but

Table VIII

Effect of Various Parameters on Average Values

Run	Description	t(hr)	$\bar{T}$ (°C)	Avg.Conv.,%	$\bar{\mu}_n$	$[\bar{C}_2]$
1.	Reference Condition: Laminar flow, Non-adiabatic (h=5) R = 0.6m T_J = 260°C T_o^J = 250°C Feed condition as per eqn.27	5	264.09	28.24	67.08	0.00516
		10	287.20	73.66	123.83	0.01788
		15	294.09	88.50	149.61	0.03003
		20	293.83	89.65	151.36	0.03702
2.	R^a = 0.4m	5	264.27	29.09	68.79	0.00497
		10	287.06	74.89	126.46	0.01782
		15	292.90	88.80	150.72	0.03002
		20	292.01	89.78	152.52	0.03684
3.	R^a = 0.8m	5	263.96	27.66	65.69	0.00524
		10	287.39	73.19	122.80	0.01786
		15	294.71	88.38	149.09	0.03000
		20	294.73	89.59	150.80	0.03706
4.	T_J^a = 250°C	5	263.49	28.03	66.71	0.00486
		10	286.44	73.67	124.19	0.01751
		15	293.12	88.55	150.22	0.02969
		20	292.64	89.73	152.13	0.03670
5.	T_o^a = 240°C	5	247.96	15.38	53.06	0.00253
		10	264.57	47.73	89.55	0.00940
		15	278.46	74.90	128.58	0.01757
		20	284.75	88.09	153.77	0.02685

Table VIII (Contd.)

6. $[W]_o^a$ = 0.13 moles/kg	5	260.25	20.76	65.07	0.00352
	10	280.21	60.04	117.87	0.01301
	15	292.05	84.16	161.87	0.02413
	20	293.84	89.17	170.77	0.03260
7. $[W]_o^a$ = 0.18 moles/kg	5	266.66	33.24	67.38	0.00638
	10	290.52	80.16	125.50	0.02108
	15	294.38	89.30	140.50	0.03305
	20	293.72	89.71	141.25	0.03892

[a]All other value as in reference condition.

are higher near equilibrium. Lower feed water concentration (Run.6, Table VIII) causes lower conversion and $\bar{\mu}_n$ initially, as the ring opening step (step.1, Table I) is slower. But as t increases, and near-equilibrium conditions are attained, conversion improves and reaches the values for the reference state, since equilibrium conversion of polycondensation reaction (step.2, Table I) is higher for lower [W]. But the effect on $\bar{\mu}_n$ is much more pronounced. This is because a shift of the equilibrium for the polycondensation step to the right hand side increases the chain length considerably. The effects of higher initial water concentration (Run.7, Table VIII), are opposite in comparison to run.6, Table VIII.

The effect of some 'computational variables' is now studied. The values of the 'slip', i.e., the non-dimensional velocity (v_z (r=R)/v_{av}) assumed at the wall (to avoid overflow and other computational problems) was changed aroung the reference value of 10^{-4} and the effects are shown in Table IX. It is observed that the results are relatively unaffected by changing the slip, but the CPU time requirement increases for lower values of slip. Similarly, instead of using the actual feed conditions at t=0, r=R, values of $[C_1]_o$, $[W]_o$, $[P_1]_o$, etc., close to equilibrium values at the wall temperature were used for the laminar flow case, to makes the equations less stiff. These values were changed to see their effect on the numerical results further down the reactor. The results were found to be quite

Table IX

Effect of Slip on Simulation Runs

Run[a]	Slip Used	Wall Temp. at 5 hr (°C)	CPU Time (for t= 5 hr)
1	10^{-4}(ref)	287.51	2 min 25 sec
2	10^{-6}	287.51	3 min 5 sec
3	10^{-3}	287.51	2 min 6 sec

[a]All runs are for laminar flow adiabatic reactor with feed condition as per eqn. 27

insensitive, but feeding in the near-equilibrium values reduced the computational time.

Fig.20 shows the effect of increasing the number of FD points on the results at t=5 hrs. The runs were made using method III, with the spacing of grid points denser near the wall (eqn.16). The figure shows very clearly that the results, particularly near the wall, converge as the number of points are increased and after about 8 points the results do not show any major discrepancy.

Methods I and III (for identical placement of FD points)were found to give the same results for various reactor conditions. This confirms that these two computer programs, one in the NAG Library and one made by us, are essentially the same. Both of them use the FD technique with GEAR's method. Method III, however, is more flexible since it permits the use of unequal spacing of FD grid points.

Some discrepancy near the wall was found between methods I and II. Table X shows the wall temperatures (which is the most sensitive variable)for the laminar flow adiabatic reactor as predicted by methods I and II, at t=5 hrs. With 9FD grid points, the results show some differences at the wall, though those at the internal positions agree well. The reason for this discrepancy lies in the fact that in method II, the wall temperature is obtained using the boundary condition alone (eqn.15). Therefore

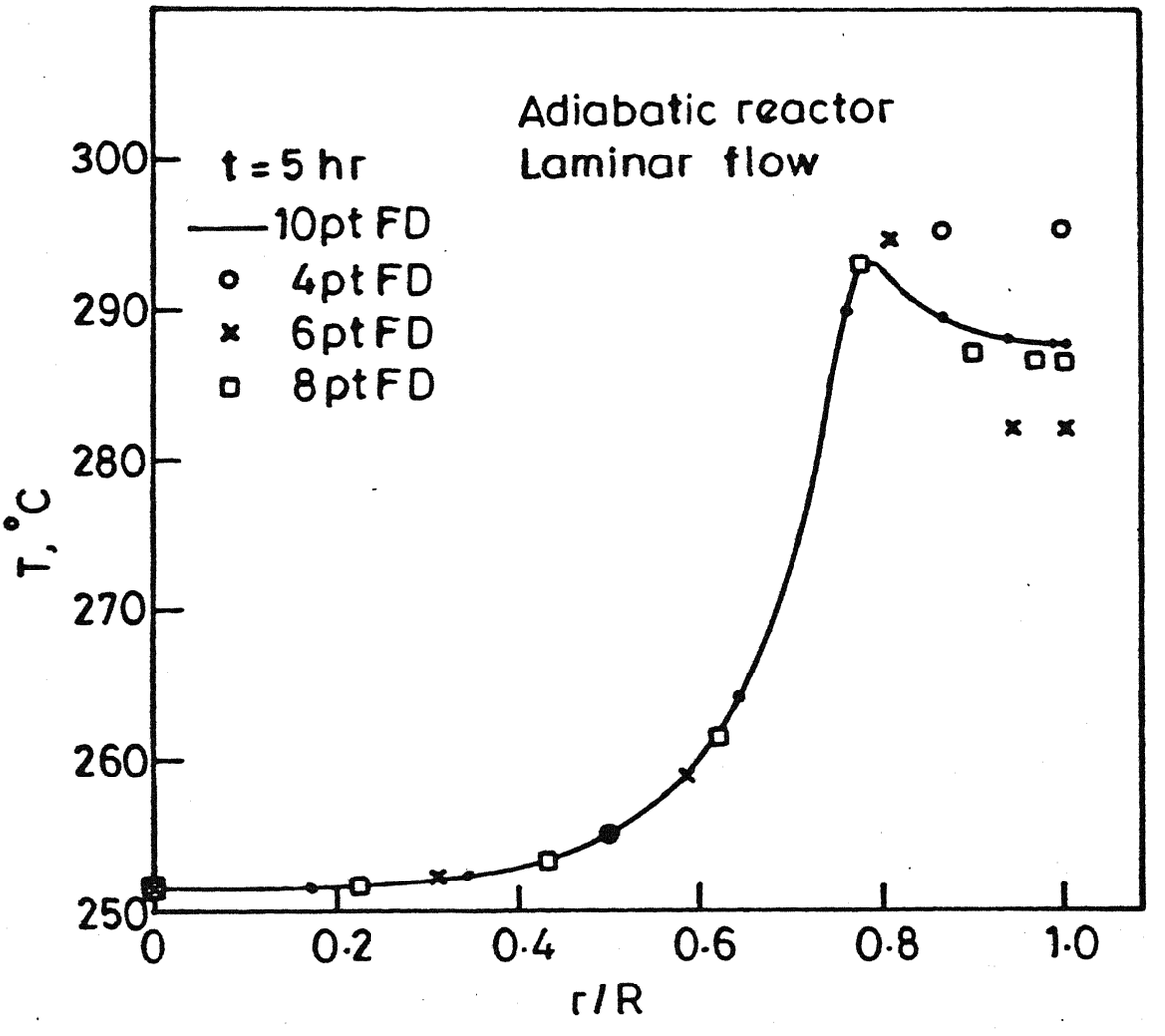


Fig. 20.

Temperature profile at t=5 hr in a laminar flow adiabatic reactor (feed conditions given in eqn. 27) for different number of grid points using finite difference (method III).

Table X

Comparison of Methods I and II

(Laminar Flow Adiabatic Reactor)

No. of FD pts., (N+1)	Temp. at the Wall (T_{N+1}), °C, at t=5hr	
	Method-I	Method-II
9	290.04	294.85
12	285.72	288.68
15	287.62	288.04
20	287.50	287.50

the ODE from the heat balance equation at the wall (r=R), and thus the heat of reaction information near that location, is not used. In method I, on the other hand, both the boundary condition and the ODE from the heat balance are used at r=R. Therefore, when the gradients are steep near the wall (for lower values of t), method I gives better predictions. Thus method I is superior to method II, particularly with regard to the results near the wall. However, as the numbers of FD points are increased, the results for both the method converge as seen in Table X. Similar results are obtained for the non-adiabatic case. However, for the non-adiabatic case of the plug flow reactor, where the temperature gradients near the wall are not so steep, the results agree well for the two methods (I & II).

The orthogonal collocation technique gives results which match those from method III for all cases of plug flow reactor (run. 1-3, Table VII) as also for the isothermal laminar flow reactor (run.4, Table VII). However, the predictions do not match well for the adiabatic (see Fig.21) and non-adiabatic cases of the laminar flow reactor (Runs. 2 & 3, Table VII). The mismatch is particularly large for lower values of t, when a steep gradient of temperature exists near the wall. The ability of our computer program for the OC method to predict good results for cases where the temperature gradients are not too steep near the wall, indicates the correctness of the program.

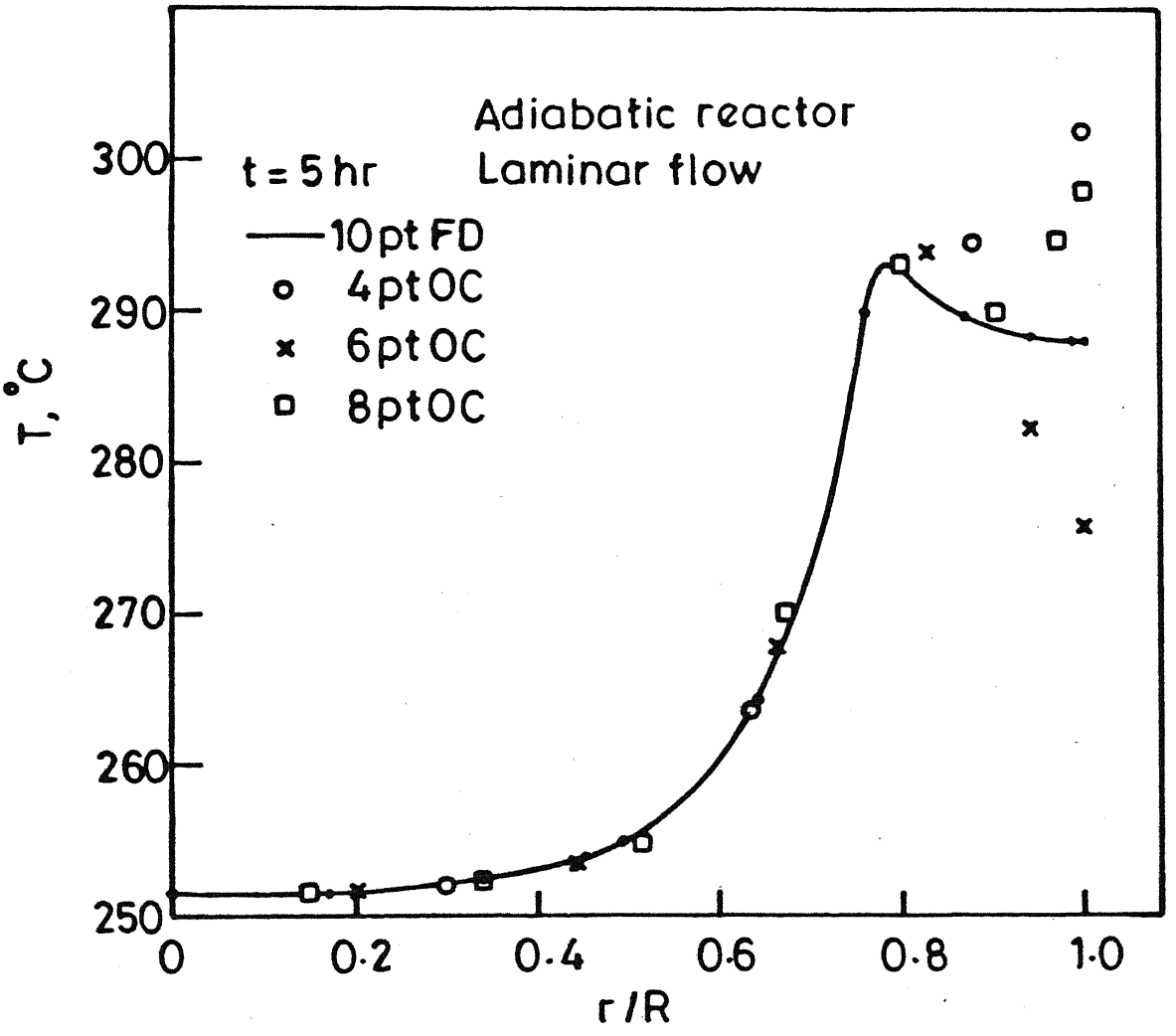


Fig. 21.

Temperature profile at t=5 hr in a laminar flow adiabatic reactor (feed condition given by eqn. 27) for different number of grid points using orthogonal collocation (method IV)

It is felt that the incorrect results obtained for the non-isothermal cases of the laminar flow reactor at lower values of t could be attributed to the fact that like method II, the OC method also does not incorporate the ODE from the heat balance at the wall. Instead, it computes the wall temperature (T_{N+1}) using all the other values of temperature, $T_1, T_2 \ldots, T_N$ (eqn.20). The OC method does not compare too well with method II either. This is due to the fact that in calculating the wall temperature (T_{N+1}), method II uses information only for the <u>two</u> adjacent points, T_N and T_{N-1} (eqn.15), whereas the OC method requires information of <u>all</u> other points (eqn.20), giving different weightages to them. Thus the prediction of the wall temperature in the OC method is likely to be sensitive to the position of the maximum (as it shifts) in the temperature profile. As t increases, the temperature maximum flattens out and the OC method is indeed found to give better results. The results for the OC method, where steep temperature gradients exist, are likely to improve on the introduction of the finite element technique.[23]

CONCLUSIONS

A detailed simulation of nylon-6 polymerization in continuous flow tubular reactors was carried out using four methods. Two types of finite difference techniques as well as the method of orthogonal collocation were used to convert PDEs into ODEs. GEAR's method was used to solve the resultant set of ODEs.

The simulation results showed the existence of significant radial gradients for temperature and concentrations for the laminar flow cases, particularly for lower values of t (average residence time). A temperature peak forms at lower t near the wall, which flattens out and simultaneously shifts towards the center further downstream. The monomer conversion and number average chain length profiles result from the temperature profiles. Higher temperature normally increases conversion and number average chain length, but it has a reverse effect where near-equilibrium conditions exist. A characteristic 'S'-shape was observed in case of plug flow in the plot of average temperature and $\bar{\mu}_n$ versus t. The plot was much flatter in the laminar flow case. The cases of adiabatic and non-adiabatic reactors both with plug and laminar flow were found not to give much different results for the average value of μ_n, which is of greater importance as far as the final product properties are concerned.

Methods I and III were found to be essentially similar with the latter having more flexibility as it could use any type of spacing of grid points. Method II was found to give poorer results near the wall particularly when the temperature gradient near the wall was steep. Both the methods I and II agreed wall as the number of grid points was increased. Method IV gave good results for all the cases except the adiabatic and non-adiabatic cases in laminar flow situation near the wall where the temperature gradients were particularly steep. Use of the finite element method is suggested to give improved results for such cases.

REFERENCES

1. H.K. Reimschuessel, J. Appl. Polym. Sci., Macromol. Revs., 12, 65 (1977).

2. K. Tai and T. Tagawa, Ind. Eng. Chem., Prod. Res. Dev., 22, 192 (1983).

3. A. Kumar and S.K. Gupta, J. Macromol. Sci., Revs. in Macromol. Chem. and Phy., C26, 183 (1986).

4. S.K. Gupta, Nylon Polymerization, in N.P. Cheremisinoff, ed., Encyclopedia of Engineering Materials, Vol.I, Properties and Synthesis, Dekker, NY, 1988.

5. S.K. Gupta and A. Kumar, Reaction Engineering of Step Growth Polymerization, Plenum, N.Y., 1987.

6. K. Tai, H. Teranishi, Y. Arai and T. Tagawa, J. Appl. Polym. Sci., 25, 77 (1980)

7. K. Tai, Y. Arai and T. Tagawa, J. Appl. Polym. Sci., 27, 731 (1982).

8. K. Nagasubramanian and H.K. Reimschuessel, J. Appl. Polym. Sci., 16, 929 (1972).

9. S. Mochizuki and N. Ito, Chem. Eng. Sci., 33, 140 (1978).

10. A. Gupta and K.S. Gandhi, Frontiers in Chem. Rxn. Eng., L.K. Doraiswamy and R.A. Mashelkar, eds., Wiley Eastern, New Delhi, 1984, pp 667-682.

11. S.K. Gupta and M. Tjahjadi, J. Appl. Polym. Sci., 33, 933 (1987).

12. A. Gupta, S.K. Gupta, K.S. Gandhi, B.V. Ankleswaria, M.H. Mehta, M.R. Padh and A.V. Soni, Recent Trends in Chem. Rxn. Eng., eds B.D. Kulkarni, R.A. Mashelkar and M.M. Sharma, Wiley Eastern, New Delhi, 1987, pp 281-297.

13. W.F.H. Naudin ten Cate, Proc. Intl. Cong. on Use of Elec. Computers in ChE, Paris, April 1973.

14. K. Nagasubramanian and H.K. Reimschuessel, J. Appl. Polym. Sci., 17, 1663 (1973).

15. K. Tai, Y. Arai and T. Tagawa, ., J. Appl. Polym. Sci., 28, 2727 (1983).

16. K. Tai and T. Tagawa, J. Appl. Polym. Sci., 27, 2791 (1982).

17. S.K. Gupta, A. Kumar, P. Tandon and C.D. Naik, Polymer, 22, 481 (1981).

18. S.K. Gupta, C.D. Naik, P. Tandon and A. Kumar, J. Appl. Polym. Sci., 26, 2153 (1981).

19. A. Kumar and S.K. Gupta, Polymer, 22, 1760 (1981).

20. J.W. Hamer and W.H. Ray, Chem. Eng. Sci., 41, 3083 (1986).

21. H. Jacobs and C. Schweigman, Proc. 5th Eur/2nd Internl. Symp. Chem. Rxn. Eng., Amsterdam, May 2-4, p.B7.1 (1972).

22. Y. Arai, K. Tai, H. Teranishi and T. Tagawa, Polymer, 22, 273 (1981).

23. B.A. Finlayson, Nonlinear Analysis in Chemical Eng., McGraw-Hill, N.Y., 1980.

24. P.E. Gill and G.F. Miller, Comp. Journal, 15, 80 (1972).

25. A.K. Ray and S.K. Gupta, Polymer, 26, 1033 (1986).

APPENDIX

```
C     ***********************************************
CCCC
      FINITE DIFFERENCE: METHOD I
C     ***********************************************
      DIMENSION COL(25),Y(150),W(150,200),RX(25)
      COMMON /OUT/H,IT,X,N
      COMMON ICOL,COL,U10,RX,UO,TW,R,TK,U70,ICOND,IVEL,DR
      EXTERNAL FCN,OUTPUT,PEDERV
      OPEN(UNIT=15,DEVICE='DSK',FILE='FD.IN')
      OPEN(UNIT=39,DEVICE='DSK',FILE='FD.OUT')
      OPEN(UNIT=6,DEVICE='DSK',FILE='ERR2.OUT')
      READ(15,*) ICOND,IVEL,NPTS,
     1           UO,TK,TW,
     2              R,XEND,
     3             U10,U20,U30,U40,U50,U60,U70
      ICOL=NPTS-1
      IF(ICOND.EQ.0) WRITE(39,17)
      IF(ICOND.EQ.1) WRITE(39,18)
      IF(ICOND.EQ.2) WRITE(39,19)
17    FORMAT(15X,50('*'),///,30X,'ISOTHERMAL REACTOR',///,15X,50('*')-
     1  //)
18    FORMAT(15X,50('*'),///,30X,'ADIABATIC REACTOR',///,15X,50('*'),
     1  //)
19    FORMAT(15X,50('*'),///,22X,'TUBULAR REACTOR WITH HEAT
     1  TRANSFER',///,15X,50('*'),//)
      WRITE(39,20) R,NPTS,U10,U40,U20,U50,U70,U30,U60
20    FORMAT(5X,'RADIUS=',F6.2,5X,'NO OF F.D. POINTS=',I3,/,
     1  5X,'MONOMER=',F8.4,5X,'1st MOMENT=',F8.4,5X,/,
     2  8X,'P(1)=',F8.4,10X,'DIMER=',F8.4,10X,'TEMP= ',F10.4,/,
     3  2X,'Oth MOMENT=',F8.4,10X,'WATER=',F8.4,/,75('*'))
      TN=TW-273.
      IF(ICOND.EQ.2) WRITE(39,21) UO,TN
21    FORMAT(/,5X,'HEAT TRANSF COEFF= ',F10.5,2X,'KCAL/(SQ M)(HR)
     1  (DEG K)',/,10X,'COOLANT TEMP= ',F10.5,//,75('*'))
      TOL=1.E-06
      IRELAB=2
      MPED=0
      X=0.0
      H=0.5
      IT=0
      DR=1.0/ICOL
      DO 11 I=1,(ICOL+1)
11    COL(I)=FLOAT(I-1)*DR
      DO 33 J=1,ICOL+1
      RX(J)=2.*(1.-(COL(J))**2.)
      IF(RX(J).EQ.0.0) RX(J)=1.E-04
      IF(IVEL.EQ.0) RX(J)=1.0
33    CONTINUE
C     INITIAL CONDITION
      DO 22 I=1,(ICOL+1)
      Y(I)=U10
      Y(ICOL+1+I)=U20
      Y(ICOL*2+2+I)=U30

      Y(ICOL*3+3+I)=U40
      Y(ICOL*4+4+I)=U50
      Y(ICOL*5+5+I)=U60
      Y(ICOL*6+6+I)=U70
22    CONTINUE
      IF(IVEL.EQ.0) GO TO 23
      Y(ICOL+1)=0.670647
      Y(ICOL*2+2)=0.000234
      Y(ICOL*3+3)=0.043585
      Y(ICOL*4+4)=8.05944
      Y(ICOL*5+5)=0.034956
      Y(ICOL*6+6)=0.116415
23    IFAIL=0
      N=(ICOL+1)*7
      IW=18+N
      CALL D02EBF(X,XEND,N,Y,TOL,IRELAB,FCN,MPED,
```

```
     1            PEDERV,OUTPUT,W,IW,IFAIL)
      WRITE(39,26) (COL(I),I=1,ICOL+1)
26    FORMAT(10X,10F8.3)
      IF(IFAIL.NE.0) GO TO 24
      STOP
24    TYPE *,IFAIL
      END
C     ***********************************************
CC    SUBROUTINE FCN
C     ***********************************************
      SUBROUTINE FCN(T,Y,F)
      DIMENSION Y(150),F(150),AO(5),EO(5),FK(5,25),
     1  BK(5,25),EK(5,25),DS(5),DH(5),AC(5),EC(5),COL(25),RHO(25)
     2  ,CP(25),RX(25),TP(25),F1(25),F2(25)
      COMMON ICOL,COL,U10,RX,UO,TW,R,TK,U70,ICOND,IVEL,DR
      DO 44 J=1,ICOL+1
44    TP(J)=Y(ICOL*6+6+J)
      TP(ICOL+2)=TP(ICOL)-2.0*DR*UO*R/TK*(TP(ICOL+1)-TW)
C     TYPE *,TP(ICOL),TP(ICOL+1),TP(ICOL+2)
      DO 16 J=1,ICOL+1
      RHO(J)=1000.0/(1.0065+0.0123*Y(J)+(TP(J)-495.)*
     3  (0.00035+0.00007*Y(J)))
      CP(J)=0.6593*Y(J)/U10+(1.-Y(J)/U10)*(486.1+0.337*TP(J)
     4  )/1000.0
16    CONTINUE
      AO(1)=5.9874E+05
      AO(2)=1.8942E+10
      AO(3)=2.8558E+09
      AO(4)=8.5778E+11
      AO(5)=2.5701E+08
      EO(1)=1.988E+04
      EO(2)=2.3271E+04
      EO(3)=2.2845E+04
      EO(4)=4.2E+04
      EO(5)=2.13E+04
      DH(1)=1.918E+03
      DH(2)=-5.9458E+03
      DH(3)=-4.0438E+03
      DH(4)=-9.6E+03
      DH(5)=-3.1691E+03
      DS(1)=-7.8846
      DS(2)=9.4374E-01
      DS(3)=-6.9487
      DS(4)=-14.52
      DS(5)=5.8265E-01
C     ..................................................
      AC(1)=4.3075E+07
      AC(2)=1.2114E+10
      AC(3)=1.6377E+10
      AC(4)=2.3307E+12
      AC(5)=3.011E+09
      EC(1)=1.8806E+04
      EC(2)=2.067E+04
      EC(3)=2.0107E+04
      EC(4)=3.74E+04
      EC(5)=2.04E+04
      UR=1.987
      DO 11 I=1,5
      DO 11 J=1,ICOL+1
      FK(I,J)=AO(I)*EXP(-EO(I)/(UR*TP(J)))+AC(I)*EXP
     1  (-EC(I)/(UR*TP(J)))*Y(ICOL*2+2+J)
      EK(I,J)=EXP(DS(I)/UR-DH(I)/(UR*TP(J)))
      BK(I,J)=FK(I,J)/EK(I,J)
11    CONTINUE
      DO 12 J=1,ICOL+1
      P2=Y(ICOL+1+J)
      P3=Y(ICOL+1+J)
      APR=(Y(ICOL*2+2+J)-2.*Y(ICOL+1+J))
      IF(APR.LT.0.0) APR=0.0
      IF(APR.LT.0.0) P2=0.0
      IF(APR.LT.0.0) P3=0.0
```

```
C
      F(J)=-FK(1,J)*Y(J)*Y(ICOL*5+5+J)+BK(1,J)*Y(ICOL+1+J)
     1   -FK(3,J)*Y(J)*Y(ICOL*2+2+J)
     2   +BK(3,J)*(Y(ICOL*2+2+J)-Y(ICOL+1+J))
      F(J)=F(J)/RX(J)
C
      F(ICOL+1+J)=FK(1,J)*Y(J)*Y(ICOL*5+5+J)-BK(1,J)*Y(ICOL+1+J)
     1   -2.*FK(2,J)*Y(ICOL+1+J)*Y(ICOL*2+2+J)
     2   +2.*BK(2,J)*Y(ICOL*5+5+J)*(Y(ICOL*2+2+J)-Y(ICOL+1+J))
     3   -FK(3,J)*Y(ICOL+1+J)*Y(J)+
     4     BK(3,J)*P2-FK(5,J)*Y(ICOL+1+J)*Y(ICOL*4+4+J)+BK(5,J)*P3
      F(ICOL+1+J)=F(ICOL+1+J)/RX(J)
C
      F(ICOL*2+2+J)=FK(1,J)*Y(J)*Y(ICOL*5+5+J)-BK(1,J)*Y(ICOL+1+J)
     1   -FK(2,J)*Y(ICOL*2+2+J)*Y(ICOL*2+2+J)+
     2     BK(2,J)*Y(ICOL*5+5+J)*(Y(ICOL*3+3+J)-Y(ICOL*2+2+J))
     3   +FK(4,J)*Y(ICOL*4+4+J)*Y(ICOL*5+5+J)-
     4     BK(4,J)*P2
      F(ICOL*2+2+J)=F(ICOL*2+2+J)/RX(J)
C
      F(ICOL*3+3+J)=FK(1,J)*Y(J)*Y(ICOL*5+5+J)-BK(1,J)*Y(ICOL+1+J)
     1   +FK(3,J)*Y(J)*Y(ICOL*2+2+J)-
     2     BK(3,J)*(Y(ICOL*2+2+J)-Y(ICOL+1+J))+2.*FK(5,J)*
     3   Y(ICOL*4+4+J)*Y(ICOL*2+2+J)-2.*BK(5,J)*
     4     APR+2.*FK(4,J)*Y(ICOL*5+5+J)*Y(ICOL*4+4+J)-2.*BK(4,J)*P2
      F(ICOL*3+3+J)=F(ICOL*3+3+J)/RX(J)
C
      F(ICOL*4+4+J)=-FK(4,J)*Y(ICOL*4+4+J)*Y(ICOL*5+5+J)+BK(4,J)*P2
     1   -FK(5,J)*Y(ICOL*4+4+J)*Y(ICOL*2+2+J)+
     2     BK(5,J)*APR
      F(ICOL*4+4+J)=F(ICOL*4+4+J)/RX(J)
C
      F(ICOL*5+5+J)=-FK(1,J)*Y(J)*Y(ICOL*5+5+J)+BK(1,J)*Y(ICOL+1+J)
     1   +FK(2,J)*Y(ICOL*2+2+J)*Y(ICOL*2+2+J)-
     2     BK(2,J)*Y(ICOL*5+5+J)*(Y(ICOL*3+3+J)-Y(ICOL*2+2+J))
     3   -FK(4,J)*Y(ICOL*4+4+J)*Y(ICOL*5+5+J)+BK(4,J)
     2     *P2
      F(ICOL*5+5+J)=F(ICOL*5+5+J)/RX(J)
C
12    CONTINUE
      DO 13 JJ=1,ICOL+1
      P2=Y(ICOL+1+JJ)
      P3=Y(ICOL+1+JJ)
      APR=(Y(ICOL*2+2+JJ)-2.*Y(ICOL+1+JJ))
      IF(APR.LT.0.0) APR=0.0
      IF(APR.LT.0.0) P2=0.0
      IF(APR.LT.0.0) P3=0.0
      F1(JJ)=-DH(1)*(FK(1,JJ)*Y(JJ)*Y(ICOL*5+5+JJ)-BK(1,JJ)*
     7   Y(ICOL+1+JJ))
     1   -DH(2)*(FK(2,JJ)*Y(ICOL*2+2+JJ)*Y(ICOL*2+2+JJ)-BK(2,JJ)*
     2   Y(ICOL*5+5+JJ)*(Y(ICOL*3+3+JJ)-Y(ICOL*2+2+JJ)))
     3   -DH(3)*(FK(3,JJ)*Y(JJ)*Y(ICOL*2+2+JJ)-BK(3,JJ)*
     4   Y(ICOL*2+2+JJ)+BK(3,JJ)*Y(ICOL+1+JJ))
     5   -DH(4)*(FK(4,JJ)*Y(ICOL*4+4+JJ)*Y(ICOL*5+5+JJ)-BK(4,JJ)*P2)
     6   -DH(5)*(FK(5,JJ)*Y(ICOL*4+4+JJ)*Y(ICOL*2+2+JJ)-BK(5,JJ)*APR)
13    CONTINUE
C
      DO 15 L=2,ICOL+1
      F2(L)=(TK/R**2.0)*((TP(L+1)-TP(L-1))/2.0/DR
     1   /COL(L)+(TP(L+1)-2.0*TP(L)+TP(L-1))/(DR**2.0))
15    CONTINUE
C     TYPE *,F1(ICOL+1),F2(ICOL+1)
      F2(1)=4.0*TK/(R**2.0)*(TP(2)-TP(1))
      DO 17 J=1,ICOL+1
      F(ICOL*6+6+J)=(F2(J)+RHO(J)*F1(J)/1000.0)
     2   /(RX(J)*RHO(J)*CP(J))
      IF(ICOND.EQ.0) F(ICOL*6+6+J)=0.0
17    CONTINUE
      RETURN
      END
C     ************************************************
```

```
C       SUBROUTINE OUTPUT
C       ************************************************
        SUBROUTINE   OUTPUT(XSOL,Y)
        DIMENSION Y(150),RHO(25),CP(25),
     1  COL(25),RY(10,25,50),TY(15,50),U(25),UU(25),U1(25),U2(25),
     2  U3(25),U4(25),U5(25),U6(25),U7(25),RX(25),TP(25)
        COMMON /OUT/H,IT,X,N
        COMMON ICOL,COL,U10,RX,U0,TW,R,TK,U70,ICOND,IVEL,DR
        TYPE *,IT
        IF(IT.EQ.0) GO TO 36
        DO 17 J=1,ICOL+1
        RY(1,J,IT)=Y(J)
        RY(2,J,IT)=Y(ICOL+1+J)
        RY(3,J,IT)=Y(ICOL*2+2+J)
        RY(4,J,IT)=Y(ICOL*3+3+J)
        RY(5,J,IT)=Y(ICOL*4+4+J)
        RY(6,J,IT)=Y(ICOL*5+5+J)
        RY(7,J,IT)=Y(ICOL*6+6+J)-273.
        RY(8,J,IT)=RY(4,J,IT)/RY(3,J,IT)
55      RY(9,J,IT)=(1.-RY(1,J,IT)/U10)*100.0
17      CONTINUE
        DO 16 J=1,ICOL+1
        RHO(J)=1000.0/(1.0065+0.0123*Y(J)+(RY(7,J,IT)+273.0-495.)*
     3  (0.00035+0.00007*Y(J)))
        CP(J)=0.6593*Y(J)/U10+(1.-Y(J)/U10)*(486.1+0.337*(RY(7,J,IT)
     4  +273.0))/1000.0
16      CONTINUE
        DO 18 J=1,ICOL+1
        U(J)=RHO(J)*RX(J)*COL(J)
        UU(J)=U(J)*CP(J)
        U1(J)=U(J)*RY(1,J,IT)
        U2(J)=U(J)*RY(2,J,IT)
        U3(J)=U(J)*RY(3,J,IT)
        U4(J)=U(J)*RY(4,J,IT)
        U5(J)=U(J)*RY(5,J,IT)
        U6(J)=U(J)*RY(6,J,IT)
        U7(J)=U(J)*(RY(7,J,IT)+273.)*CP(J)
18      CONTINUE
        IPT=ICOL+1
        IFAIL1=0
        CALL D01GAF(COL,U,IPT,SUM,ER,IFAIL1)
        CALL D01GAF(COL,UU,IPT,SUMM,ERR,IFAIL1)
        CALL D01GAF(COL,U1,IPT,SUM1,ER1,IFAIL1)
        CALL D01GAF(COL,U2,IPT,SUM2,ER2,IFAIL1)
        CALL D01GAF(COL,U3,IPT,SUM3,ER3,IFAIL1)
        CALL D01GAF(COL,U4,IPT,SUM4,ER4,IFAIL1)
        CALL D01GAF(COL,U5,IPT,SUM5,ER5,IFAIL1)
        CALL D01GAF(COL,U6,IPT,SUM6,ER6,IFAIL1)
        CALL D01GAF(COL,U7,IPT,SUM7,ER7,IFAIL1)
        IF(IFAIL.NE.0) GO TO 99
        U1AV=SUM1/SUM
        TY(10,IT)=(1.-U1AV/U10)*100.
        TY(11,IT)=SUM4/SUM3
        TY(12,IT)=SUM6/SUM
        TY(13,IT)=SUM7/SUMM-273.
        TY(14,IT)=SUM5/SUM
        IF(XSOL.EQ.0.) GO TO 35
        WRITE(39,22) XSOL
22      FORMAT(11X,'TIME:',F5.2,2X,'Hr.',//)
        WRITE(39,23)  (RY(1,I,IT),I=1,ICOL+1)
        WRITE(39,39) (RY(2,I,IT),I=1,ICOL+1),(RY(3,I,IT),I=1,ICOL+1)
     1  ,(RY(4,I,IT),I=1,
     2  ICOL+1),(RY(5,I,IT),I=1,ICOL+1),(RY(6,I,IT),I=1,ICOL+1)
39      FORMAT(10X,'P(1) :',10F10.6,/,5X,'0th MOMENT:',10F10.6,/,
     1  5X,'1st MOMENT:',10F10.6,/,9X,'DIMER :',10F10.6,/,
     2  9X,'WATER :',10F10.6 )
23      FORMAT(8X,'MONOMER:',10F10.6)
        WRITE(39,24) (RY(7,I,IT),I=1,ICOL+1)
24      FORMAT(11X,'TEMP:',10F10.4)
        WRITE(39,26) (RY(8,I,IT),I=1,ICOL+1)
26      FORMAT(/,5X,'CHAIN LTH.: ',10F10.3)
```

```
      WRITE(39,29) (RY(9,I,IT),I=1,ICOL+1)
29    FORMAT(4X,'CONVERSION :',10F10.3)
      WRITE(39,47) TY(10,IT),TY(13,IT),TY(11,IT),TY(14,IT),TY(12,IT)
47    FORMAT(/,7X,'AV CONV= ',F10.6,5X,'AV TEMP= ',F10.4,/,
     1  2X,'AV CHAIN LTH= ',F10.6,4X,'AV DIMER= ',F10.6,/,
     2  6X,'AV WATER= ',F10.6)
      WRITE(39,27)
27    FORMAT(75('.'))
36    IT=IT+1
35    XSOL=X+FLOAT(IT)*H
      RETURN
99    TYPE *,IFAIL1
      END
C     ************************************************
      SUBROUTINE PEDERV(T,Y,PW)
      DIMENSION Y(150),PW(70,70)
      RETURN
      END
```

```
C     ***********************************************
CCCCC
      FINITE DIFFERENCE: METHOD II
CCCCC
      ***********************************************
      DIMENSION COL(25),Y(150),W(150,200),RX(25)
      COMMON /OUT/H,IT,X,N
      COMMON ICOL,COL,U10,RX,UO,TW,R,TK,U70,ICOND,IVEL,DR
      EXTERNAL FCN,OUTPUT,PEDERV
      OPEN(UNIT=15,DEVICE='DSK',FILE='FD1.IN')
      OPEN(UNIT=39,DEVICE='DSK',FILE='FD1.OUT')
      OPEN(UNIT=6,DEVICE='DSK',FILE='ERR3.OUT')
      READ(15,*) ICOND,IVEL,NPTS,
     1           UO,TK,TW,
     2             R,XEND,
     3            U10,U20,U30,U40,U50,U60,U70
      ICOL=NPTS-1
      IF(ICOND.EQ.0) WRITE(39,17)
      IF(ICOND.EQ.1) WRITE(39,18)
      IF(ICOND.EQ.2) WRITE(39,19)
17    FORMAT(15X,50('*'),///,30X,'ISOTHERMAL REACTOR',///,15X,50('*').
     1  //)
18    FORMAT(15X,50('*'),///,30X,'ADIABATIC REACTOR',///,15X,50('*'),
     1  //)
19    FORMAT(15X,50('*'),///,22X,'TUBULAR REACTOR WITH HEAT
     1  TRANSFER',///,15X,50('*'),//)
      WRITE(39,20) R,NPTS,U10,U40,U20,U50,U70,U30,U60
20    FORMAT(5X,'RADIUS=',F6.2,5X,'NO OF F.D. POINTS=',I3,/,
     1  5X,'MONOMER=',F8.4,5X,'1st MOMENT=',F8.4,5X,/,
     2  8X,'P(1)=',F8.4,10X,'DIMER=',F8.4,10X,'TEMP= ',F10.4,/,
     3  2X,'0th MOMENT=',F8.4,10X,'WATER=',F8.4,/,75('*'))
      TN=TW-273.
      IF(ICOND.EQ.2) WRITE(39,21) UO,TN
21    FORMAT(/,5X,'HEAT TRANSF COEFF= ',F10.5,2X,'KCAL/(SQ M)(HR)
     1  (DEG K)',/,10X,'COOLANT TEMP= ',F10.5,//,75('*'))
      TOL=1.E-06
      IRELAB=2
      MPED=0
      X=0.0
      H=0.5
      IT=0
      DR=1.0/ICOL
      DO 11 I=1,(ICOL+1)
11    COL(I)=FLOAT(I-1)*DR
      DO 33 J=1,ICOL+1
      RX(J)=2.*(1.-(COL(J))**2.)
      IF(RX(J).EQ.0.0) RX(J)=1.E-04
      IF(IVEL.EQ.0) RX(J)=1.0
33    CONTINUE
C     INITIAL CONDITION
      DO 22 I=1,(ICOL+1)
      Y(I)=U10
      Y(ICOL+1+I)=U20
      Y(ICOL*2+2+I)=U30

      Y(ICOL*3+3+I)=U40
      Y(ICOL*4+4+I)=U50
      Y(ICOL*5+5+I)=U60
      Y(ICOL*6+6+I)=U70
22    CONTINUE
      IF(IVEL.EQ.0) GO TO 23
      Y(ICOL+1)=0.670647
      Y(ICOL*2+2)=0.000234
      Y(ICOL*3+3)=0.043585
      Y(ICOL*4+4)=8.05944
      Y(ICOL*5+5)=0.034956
      Y(ICOL*6+6)=0.116415
23    IFAIL=0
      N=(ICOL+1)*7-2
      IW=18+N
      CALL D02EBF(X,XEND,N,Y,TOL,IRELAB,FCN,MPED,
```

```
     1            PEDERV,OUTPUT,W,IW,IFAIL)
      WRITE(39,26) (COL(I),I=1,ICOL+1)
26    FORMAT(10X,20F8.3)
      IF(IFAIL.NE.0) GO TO 24
      STOP
24    TYPE *,IFAIL
      END
C     ************************************************
C     SUBROUTINE FCN
C     ************************************************
      SUBROUTINE FCN(T,Y,F)
      DIMENSION Y(150),F(150),AO(5),EO(5),FK(5,25),
     1  BK(5,25),EK(5,25),DS(5),DH(5),AC(5),EC(5),COL(25),RHO(25)
     2  ,CP(25),RX(25),TP(25),F1(25),F2(25)
      COMMON ICOL,COL,U10,RX,UO,TW,R,TK,U70,ICOND,IVEL,DR
      DO 44 J=1,ICOL-1
44    TP(J+1)=Y(ICOL*6+6+J)
      TP(1)=(4.0*TP(2)-TP(3))/3.0
      TP(ICOL+1)=(UO*TW-(TK/(2.0*DR*R))*(TP(ICOL-1)-4.0
     1  *TP(ICOL)))/(UO+(3.0*TK)/(2.0*DR*R))
C     TYPE *,TP(ICOL),TP(ICOL+1),TP(ICOL+2)
      DO 16 J=1,ICOL+1
      RHO(J)=1000.0/(1.0065+0.0123*Y(J)+(TP(J)-495.)*
     3  (0.00035+0.00007*Y(J)))
      CP(J)=0.6593*Y(J)/U10+(1.-Y(J)/U10)*(486.1+0.337*TP(J)
     4  )/1000.0
16    CONTINUE
      AO(1)=5.9874E+05
      AO(2)=1.8942E+10
      AO(3)=2.8558E+09
      AO(4)=8.5778E+11
      AO(5)=2.5701E+08
      EO(1)=1.988E+04
      EO(2)=2.3271E+04
      EO(3)=2.2845E+04
      EO(4)=4.2E+04
      EO(5)=2.13E+04
      DH(1)=1.918E+03
      DH(2)=-5.9458E+03
      DH(3)=-4.0438E+03
      DH(4)=-9.6E+03
      DH(5)=-3.1691E+03
      DS(1)=-7.8846
      DS(2)=9.4374E-01
      DS(3)=-6.9487
      DS(4)=-14.52
      DS(5)=5.8265E-01
C     ................................................
      AC(1)=4.3075E+07
      AC(2)=1.2114E+10
      AC(3)=1.6377E+10
      AC(4)=2.3307E+12
      AC(5)=3.011E+09
      EC(1)=1.8806E+04
      EC(2)=2.067E+04
      EC(3)=2.0107E+04
      EC(4)=3.74E+04
      EC(5)=2.04E+04
      UR=1.987
      DO 11 I=1,5
      DO 11 J=1,ICOL+1
      FK(I,J)=AO(I)*EXP(-EO(I)/(UR*TP(J)))+AC(I)*EXP
     1  (-EC(I)/(UR*TP(J)))*Y(ICOL*2+2+J)
      EK(I,J)=EXP(DS(I)/UR-DH(I)/(UR*TP(J)))
      BK(I,J)=FK(I,J)/EK(1,J)
11    CONTINUE
      DO 12 J=1,ICOL+1
      P2=Y(ICOL+1+J)
      P3=Y(ICOL+1+J)
      APR=(Y(ICOL*2+2+J)-2.*Y(ICOL+1+J))
      IF(APR.LT.0.0) APR=0.0
```

```
      IF(APR.LT.0.0) P2=0.0
      IF(APR.LT.0.0) P3=0.0
C     .....
      F(J)=-FK(1,J)*Y(J)*Y(ICOL*5+5+J)+BK(1,J)*Y(ICOL+1+J)
     1   -FK(3,J)*Y(J)*Y(ICOL*2+2+J)
     2   +BK(3,J)*(Y(ICOL*2+2+J)-Y(ICOL+1+J))
      F(J)=F(J)/RX(J)
C     .....
      F(ICOL+1+J)=FK(1,J)*Y(J)*Y(ICOL*5+5+J)-BK(1,J)*Y(ICOL+1+J)
     1   -2.*FK(2,J)*Y(ICOL+1+J)*Y(ICOL*2+2+J)
     2   +2.*BK(2,J)*Y(ICOL*5+5+J)*(Y(ICOL*2+2+J)-Y(ICOL+1+J))
     3   -FK(3,J)*Y(ICOL+1+J)*Y(J)+
     4     BK(3,J)*P2-FK(5,J)*Y(ICOL+1+J)*Y(ICOL*4+4+J)+BK(5,J)*P3
      F(ICOL+1+J)=F(ICOL+1+J)/RX(J)
C     .....
      F(ICOL*2+2+J)=FK(1,J)*Y(J)*Y(ICOL*5+5+J)-BK(1,J)*Y(ICOL+1+J)
     1   -FK(2,J)*Y(ICOL*2+2+J)*Y(ICOL*2+2+J)+
     2     BK(2,J)*Y(ICOL*5+5+J)*(Y(ICOL*3+3+J)-Y(ICOL*2+2+J))
     3   +FK(4,J)*Y(ICOL*4+4+J)*Y(ICOL*5+5+J)-
     4     BK(4,J)*P2
      F(ICOL*2+2+J)=F(ICOL*2+2+J)/RX(J)
C     .....
      F(ICOL*3+3+J)=FK(1,J)*Y(J)*Y(ICOL*5+5+J)-BK(1,J)*Y(ICOL+1+J)
     1   +FK(3,J)*Y(J)*Y(ICOL*2+2+J)-
     2     BK(3,J)*(Y(ICOL*2+2+J)-Y(ICOL+1+J))+2.*FK(5,J)*
     3   Y(ICOL*4+4+J)*Y(ICOL*2+2+J)-2.*BK(5,J)*
     4     APR+2.*FK(4,J)*Y(ICOL*5+5+J)*Y(ICOL*4+4+J)-2.*BK(4,J)*P2
      F(ICOL*3+3+J)=F(ICOL*3+3+J)/RX(J)
C     .....
      F(ICOL*4+4+J)=-FK(4,J)*Y(ICOL*4+4+J)*Y(ICOL*5+5+J)+BK(4,J)*P2
     1   -FK(5,J)*Y(ICOL*4+4+J)*Y(ICOL*2+2+J)+
     2     BK(5,J)*APR
      F(ICOL*4+4+J)=F(ICOL*4+4+J)/RX(J)
C     .....
      F(ICOL*5+5+J)=-FK(1,J)*Y(J)*Y(ICOL*5+5+J)+BK(1,J)*Y(ICOL+1+J)
     1   +FK(2,J)*Y(ICOL*2+2+J)*Y(ICOL*2+2+J)-
     2     BK(2,J)*Y(ICOL*5+5+J)*(Y(ICOL*3+3+J)-Y(ICOL*2+2+J))
     3   -FK(4,J)*Y(ICOL*4+4+J)*Y(ICOL*5+5+J)+BK(4,J)
     2     *P2
      F(ICOL*5+5+J)=F(ICOL*5+5+J)/RX(J)
C     .....
12    CONTINUE
      DO 13 JJ=2,ICOL
      P2=Y(ICOL+1+JJ)
      P3=Y(ICOL+1+JJ)
      APR=(Y(ICOL*2+2+JJ)-2.*Y(ICOL+1+JJ))
      IF(APR.LT.0.0) APR=0.0
      IF(APR.LT.0.0) P2=0.0
      IF(APR.LT.0.0) P3=0.0
      F1(JJ)=-DH(1)*(FK(1,JJ)*Y(JJ)*Y(ICOL*5+5+JJ)-BK(1,JJ)*
     7   Y(ICOL+1+JJ))
     1   -DH(2)*(FK(2,JJ)*Y(ICOL*2+2+JJ)*Y(ICOL*2+2+JJ)-BK(2,JJ)*
     2   Y(ICOL*5+5+JJ)*(Y(ICOL*3+3+JJ)-Y(ICOL*2+2+JJ)))
     3   -DH(3)*(FK(3,JJ)*Y(JJ)*Y(ICOL*2+2+JJ)-BK(3,JJ)*
     4   Y(ICOL*2+2+JJ)+BK(3,JJ)*Y(ICOL+1+JJ))
     5   -DH(4)*(FK(4,JJ)*Y(ICOL*4+4+JJ)*Y(ICOL*5+5+JJ)-BK(4,JJ)*P2)
     6   -DH(5)*(FK(5,JJ)*Y(ICOL*4+4+JJ)*Y(ICOL*2+2+JJ)-BK(5,JJ)*APR)
13    CONTINUE
C     .....
      DO 15 L=2,ICOL
      F2(L)=(TK/R**2.0)*((TP(L+1)-TP(L-1))/2.0/DR
     1   /COL(L)+(TP(L+1)-2.0*TP(L)+TP(L-1))/(DR**2.0))
15    CONTINUE
C     TYPE *,F1(ICOL+1),F2(ICOL+1)
      DO 17 J=1,ICOL-1
      F(ICOL*6+6+J)=(F2(J+1)+RHO(J+1)*F1(J+1)/1000.0)
     2   /(RX(J+1)*RHO(J+1)*CP(J+1))
      IF(ICOND.EQ.0) F(ICOL*6+6+J)=0.0
17    CONTINUE
      RETURN
      END
```

```
C     ******************************************************
CC    SUBROUTINE OUTPUT
CC    ******************************************************
      SUBROUTINE   OUTPUT(XSOL,Y)
      DIMENSION Y(150),RHO(25),CP(25),
     1  COL(25),RY(10,25,50),TY(15,50),U(25),UU(25),U1(25),U2(25),
     2  U3(25),U4(25),U5(25),U6(25),U7(25),RX(25),TP(25)
      COMMON /OUT/H,IT,X,N
      COMMON ICOL,COL,U10,RX,UO,TW,R,TK,U70,ICOND,IVEL,DR
      TYPE *,IT
      IF(IT.EQ.0) GO TO 36
      DO 45 J=1,ICOL-1
45    TP(J+1)=Y(ICOL*6+6+J)
      TP(1)=(4.0*TP(2)-TP(3))/3.0
      TP(ICOL+1)=(UO*TW-(TK/(2.0*DR*R))*(TP(ICOL-1)-4.0
     1  *TP(ICOL)))/(UO+(3.0*TK)/(2.0*DR*R))
      DO 17 J=1,ICOL+1
      RY(1,J,IT)=Y(J)
      RY(2,J,IT)=Y(ICOL+1+J)
      RY(3,J,IT)=Y(ICOL*2+2+J)
      RY(4,J,IT)=Y(ICOL*3+3+J)
      RY(5,J,IT)=Y(ICOL*4+4+J)
      RY(6,J,IT)=Y(ICOL*5+5+J)
      RY(7,J,IT)=TP(J)-273.
      RY(8,J,IT)=RY(4,J,IT)/RY(3,J,IT)
55    RY(9,J,IT)=(1.-RY(1,J,IT)/U10)*100.0
17    CONTINUE
      DO 16 J=1,ICOL+1
      RHO(J)=1000.0/(1.0065+0.0123*Y(J)+(RY(7,J,IT)+273.0-495.)*
     3  (0.00035+0.00007*Y(J)))
      CP(J)=0.6593*Y(J)/U10+(1.-Y(J)/U10)*(486.1+0.337*(RY(7,J,IT)
     4  +273.0))/1000.0
16    CONTINUE
      DO 18 J=1,ICOL+1
      U(J)=RHO(J)*RX(J)*COL(J)
      UU(J)=U(J)*CP(J)
      U1(J)=U(J)*RY(1,J,IT)
      U2(J)=U(J)*RY(2,J,IT)
      U3(J)=U(J)*RY(3,J,IT)
      U4(J)=U(J)*RY(4,J,IT)
      U5(J)=U(J)*RY(5,J,IT)
      U6(J)=U(J)*RY(6,J,IT)
      U7(J)=U(J)*(RY(7,J,IT)+273.)*CP(J)
18    CONTINUE
      IPT=ICOL+1
      IFAIL1=0
      CALL D01GAF(COL,U,IPT,SUM,ER,IFAIL1)
      CALL D01GAF(COL,UU,IPT,SUMM,ERR,IFAIL1)
      CALL D01GAF(COL,U1,IPT,SUM1,ER1,IFAIL1)
      CALL D01GAF(COL,U2,IPT,SUM2,ER2,IFAIL1)
      CALL D01GAF(COL,U3,IPT,SUM3,ER3,IFAIL1)
      CALL D01GAF(COL,U4,IPT,SUM4,ER4,IFAIL1)
      CALL D01GAF(COL,U5,IPT,SUM5,ER5,IFAIL1)
      CALL D01GAF(COL,U6,IPT,SUM6,ER6,IFAIL1)
      CALL D01GAF(COL,U7,IPT,SUM7,ER7,IFAIL1)
      IF(IFAIL.NE.0) GO TO 99
      U1AV=SUM1/SUM
      TY(10,IT)=(1.-U1AV/U10)*100.
      TY(11,IT)=SUM4/SUM3
      TY(12,IT)=SUM6/SUM
      TY(13,IT)=SUM7/SUMM-273.
      TY(14,IT)=SUM5/SUM
      IF(XSOL.EQ.0.) GO TO 35
      WRITE(39,22) XSOL
22    FORMAT(11X,'TIME:',F5.2,2X,'Hr.',//)
      WRITE(39,23)  (RY(1,I,IT),I=1,ICOL+1)
      WRITE(39,39) (RY(2,I,IT),I=1,ICOL+1),(RY(3,I,IT),I=1,ICOL+1)
     1  ,(RY(4,I,IT),I=1,
     2  ICOL+1),(RY(5,I,IT),I=1,ICOL+1),(RY(6,I,IT),I=1,ICOL+1)
39    FORMAT(10X,'P(1) :',20F10.6,/,5X,'0th MOMENT:',20F10.6,/,
     1  5X,'1st MOMENT:',20F10.6,/,9X,'DIMER :',20F10.6,/,
```

```
     2  9X,'WATER :',20F10.6 )
23   FORMAT(8X,'MONOMER:',20F10.6)
     WRITE(39,24) (RY(7,I,IT),I=1,ICOL+1)
24   FORMAT(11X,'TEMP:',20F10.4)
     WRITE(39,26) (RY(8,I,IT),I=1,ICOL+1)
26   FORMAT(/,5X,'CHAIN LTH.:  ',20F10.3)
     WRITE(39,29) (RY(9,I,IT),I=1,ICOL+1)
29   FORMAT(4X,'CONVERSION :',20F10.3)
     WRITE(39,47) TY(10,IT),TY(13,IT),TY(11,IT),TY(14,IT),TY(12,IT)
47   FORMAT(/,7X,'AV CONV= ',F10.6,5X,'AV TEMP= ',F10.4,/,
     1  2X,'AV CHAIN LTH= ',F10.6,4X,'AV DIMER= ',F10.6,/,
     2  6X,'AV WATER= ',F10.6)
     WRITE(39,27)
27   FORMAT(75('.'))
36   IT=IT+1
35   XSOL=X+FLOAT(IT)*H
     RETURN
99   TYPE *,IFAIL1
     END
C    ************************************************
     SUBROUTINE PEDERV(T,Y,PW)
     DIMENSION Y(150),PW(70,70)
     RETURN
     END
```

```
C     ************************************************
C
C     FINITE DIFF: METHOD III
C
C     ************************************************
C     TUBULAR REACTOR SIMULATION
C     ************************************************
C     MAIN PROGRAM
C     ************************************************
      DIMENSION U(7,50),UOUT(7,20),WORK(5000),X(50),XOUT(20)
     1   ,CMU(15),CONV(15),TEMP(15,15),Y(15),YY(15),Y1(15),Y2(15),
     2   Y3(15),Y4(15),Y5(15),Y6(15),Y7(15),RY(25,25,100),TY(25,100)
     3   ,TX(100),H(15)
      EXTERNAL BNDY,PDEF
      OPEN(UNIT=25,DEVICE='DSK',FILE='NYL.OUT')
      OPEN(UNIT=15,DEVICE='DSK',FILE='NYL.IN')
      OPEN(UNIT=55,DEVICE='DSK',FILE='PLOT.OUT')
      NPDE=7
      READ(15,*) ICOND,IVEL,
     1           UO,BB,TM,
     2           B,NPTS,L,
     3           U10,U20,U30,U40,U50,U60,U70
      IWK=5000
      IF(ICOND.EQ.0) WRITE(25,33)
      IF(ICOND.EQ.1) WRITE(25,59)
      IF(ICOND.EQ.2) WRITE(25,69)
33    FORMAT(15X,50('*'),///,30X,'ISOTHERMAL REACTOR',///,15X,50('*'),
     1   //)
59    FORMAT(15X,50('*'),///,30X,'ADIABATIC REACTOR',///,15X,50('*'),
     1   //)
69    FORMAT(15X,50('*'),///,22X,'TUBULAR REACTOR WITH HEAT
     1   TRANSFER',///,15X,50('*'),//)
      WRITE(25,37) B,NPTS,U10,U40,U20,U50,U70,U30,U60
37    FORMAT(5X,'RADIUS=',F6.2,5X,'NO OF MESH POINTS=',I3,/,
     1   5X,'MONOMER=',F8.4,5X,'1st MOMENT=',F8.4,5X,/,
     2   8X,'P(1)=',F8.4,10X,'DIMER=',F8.4,10X,'TEMP= ',F10.4,/,
     3   2X,'0th MOMENT=',F8.4,10X,'WATER=',F8.4,/,75('*'))
      TN=TM-273.
      IF(ICOND.EQ.2) WRITE(25,53) UO,TN
53    FORMAT(/,5X,'HEAT TRANSF COEFF= ',F10.5,2X,'KCAL/(SQ M)(HR)
     1   (DEG K)',/,10X,'COOLANT TEMP= ',F10.5,//,75('*'))
      M=1
      A=0.
      ACC=0.5E-06
      IMESH=4
C     INITIAL CONDITION
      DO 20 I=1,NPTS
      U(1,I)=U10
      U(2,I)=U20
      U(3,I)=U30
      U(4,I)=U40
      U(5,I)=U50
      U(6,I)=U60
      U(7,I)=U70
20    CONTINUE
C     GO TO 18
      IF(IVEL.EQ.0)  GO TO 18
      U(1,NPTS)=0.670647
      U(2,NPTS)=0.000234
      U(3,NPTS)=0.043585
      U(4,NPTS)=8.05944
      U(5,NPTS)=0.034956
      U(6,NPTS)=0.116415
      U(7,NPTS)=TM
C     TYPE *, ((U(I,J),J=1,NPTS),I=1,7,6)
18    TS=0.
      IND=0
      DO 60 IT=1,L
      TOUT=FLOAT(IT)/2.
      TX(IT)=TOUT
      IFAIL=1
```

```
      CALL D03PBF(NPDE,M,PDEF,BNDY,A,B,TS,TOUT,U,NPTS,IMESH,
     1   X,ACC,WORK,IWK,IND,IFAIL)
      IF(IFAIL.NE.0) GO TO 100
      DO 13 J=1,NPTS
      RHO=1000.0/(1.0065+0.0123*U(1,J)+(U(7,J)-495.)*(0.00035+
     1   0.00007*U(1,J)))
      CP=0.6593*U(1,J)/U10+(1.-U(1,J)/U10)*(486.1+0.337*U(7,J))
     2  /1000.
      IF(IT.NE.10) GO TO 65
C     WRITE(24,*) RHO,CP
65    IF(IVEL.NE.1) GO TO 90
      RHO=RHO*(1.-(X(J)**2./B**2.))
90    Y(J)=RHO*X(J)
      Y1(J)=Y(J)*U(1,J)
      Y2(J)=Y(J)*U(2,J)
      Y3(J)=Y(J)*U(3,J)
      Y4(J)=Y(J)*U(4,J)
      Y5(J)=Y(J)*U(5,J)
      Y6(J)=Y(J)*U(6,J)
      Y7(J)=Y(J)*CP*U(7,J)
      YY(J)=Y(J)*CP
      CONV(J)=(1.-U(1,J)/U10)*100.
      CMU(J)=U(4,J)/U(3,J)
      TEMP(7,J)=U(7,J)-273.
13    CONTINUE
      GO TO 695
C     ***********************************************
      DO 63 J=1,(NPTS-1)
      H(J)=X(J+1)-X(J)
      UAV=UAV+0.5*H(J)*(Y(J)+Y(J+1))
      UUAV=UUAV+0.5*H(J)*(YY(J)+YY(J+1))
      U1=U1+0.5*H(J)*(Y1(J)+Y1(J+1))
      U2=U2+0.5*H(J)*(Y2(J)+Y2(J+1))
      U3=U3+0.5*H(J)*(Y3(J)+Y3(J+1))
      U4=U4+0.5*H(J)*(Y4(J)+Y4(J+1))
      U5=U5+H(J)*(Y5(J)+Y5(J+1))
      U6=U6+0.5*H(J)*(Y6(J)+Y6(J+1))
      U7=U7+0.5*H(J)*(Y7(J)+Y7(J+1))
63    CONTINUE
C     ***********************************************
C
      GO TO 64
695   IFAIL1=0
      CALL D01GAF(X,Y,NPTS,UAV,ER,IFAIL1)
      CALL D01GAF(X,YY,NPTS,UUAV,ERR,IFAIL1)
      CALL D01GAF(X,Y1,NPTS,U1,ER1,IFAIL1)
      CALL D01GAF(X,Y2,NPTS,U2,ER2,IFAIL1)
      CALL D01GAF(X,Y3,NPTS,U3,ER3,IFAIL1)
      CALL D01GAF(X,Y4,NPTS,U4,ER4,IFAIL1)
      CALL D01GAF(X,Y5,NPTS,U5,ER5,IFAIL1)
      CALL D01GAF(X,Y6,NPTS,U6,ER6,IFAIL1)
      CALL D01GAF(X,Y7,NPTS,U7,ER7,IFAIL1)
      IF(IFAIL1.NE.0) GO TO 99
64    U1AV=U1/UAV
      U2AV=U2/UAV
      U3AV=U3/UAV
      U4AV=U4/UAV
      U5AV=U5/UAV
      U6AV=U6/UAV
      U7AV=U7/UUAV
      U7AVC=U7AV-273.
      CONAV=(1.-U1AV/U10)*100.
      XMU=U4AV/U3AV
C     ..................................................
CCC   RY(VARIABLE NO.,MESH PT NO., IT)
CCC   TY(VARIABLE NO., IT)
CCC   TY(I,J) : I: 10=AV CONV ; 11=AV CHAIN LENGTH
CCC                12=AV WATER; 13=AV  TEMP(C)
CCC                14=AV DIMER
C     ..................................................
      DO 28 J=1,NPTS
      RY(7,J,IT)=TEMP(7,J)
```

```
      RY(8,J,IT)=CMU(J)
      RY(9,J,IT)=CONV(J)
      DO 28 I=1,6
      RY(I,J,IT)=U(I,J)
28    CONTINUE
      TY(10,IT)=CONAV
      TY(11,IT)=XMU
      TY(12,IT)=U6AV
      TY(13,IT)=U7AVC
      TY(14,IT)=U5AV
C     IF(NPTS.EQ.12) GO TO 691
C     ****************************************
      WRITE(25,22) TOUT
22    FORMAT(11X,'TIME:',F5.2,2X,'Hr.',//)
      WRITE(25,23)  (U(1,I),I=1,NPTS)
      WRITE(25,39) (U(2,I),I=1,NPTS),(U(3,I),I=1,NPTS),(U(4,I),I=1,
     1  NPTS),(U(5,I),I=1,NPTS),(U(6,I),I=1,NPTS)
39    FORMAT(10X,'P(1) :',10F10.6,/,5X,'0th MOMENT:',10F10.6,/,
     1  5X,'1st MOMENT:',10F10.6,/,9X,'DIMER :',10F10.6,/,
     2  9X,'WATER :',10F10.6 )
23    FORMAT(8X,'MONOMER:',10F10.6)
      WRITE(25,24) (TEMP(7,I),I=1,NPTS)
24    FORMAT(11X,'TEMP:',10F10.4)
      WRITE(25,26) ( CMU(I),I=1,NPTS)
26    FORMAT(/,5X,'CHAIN LTH.: ',10F10.3)
      WRITE(25,29) (CONV(I),I=1,NPTS)
29    FORMAT(4X,'CONVERSION :',10F10.3)
      WRITE(25,47) CONAV,U7AVC,XMU,U5AV,U6AV
47    FORMAT(/,7X,'AV CONV= ',F10.6,5X,'AV TEMP= ',F10.4,/,
     1  2X,'AV CHAIN LTH= ',F10.6,4X,'AV DIMER= ',F10.6,/,
     2  6X,'AV WATER= ',F10.6)
      WRITE(25,27)
27    FORMAT(75('.'))
      TYPE *,TOUT,(U(J,1),J=1,7,6)
      GO TO 60
C     ********************************************
691   WRITE(25,101) TOUT
101   FORMAT(11X,'TIME:',F5.2,2X,'Hr.',//)
      WRITE(25,102)  (U(1,I),I=1,NPTS)
      WRITE(25,103) (U(2,I),I=1,NPTS),(U(3,I),I=1,NPTS),(U(4,I),I=1,
     1  NPTS),(U(5,I),I=1,NPTS),(U(6,I),I=1,NPTS)
103   FORMAT(10X,'P(1) :',12F10.6,/,5X,'0th MOMENT:',12F10.6,/,
     1  5X,'1st MOMENT:',12F10.6,/,9X,'DIMER :',12F10.6,/,
     2  9X,'WATER :',12F10.6 )
102   FORMAT(8X,'MONOMER:',12F10.6)
      WRITE(25,104) (TEMP(7,I),I=1,NPTS)
104   FORMAT(11X,'TEMP:',12F10.4)
      WRITE(25,105) ( CMU(I),I=1,NPTS)
105   FORMAT(/,5X,'CHAIN LTH.: ',12F10.3)
      WRITE(25,107) (CONV(I),I=1,NPTS)
107   FORMAT(4X,'CONVERSION :',12F10.3)
      WRITE(25,106) CONAV,U7AVC,XMU,U5AV,U6AV
106   FORMAT(/,7X,'AV CONV= ',F10.6,5X,'AV TEMP= ',F10.4,/,
     1  2X,'AV CHAIN LTH= ',F10.6,4X,'AV DIMER= ',F10.6,/,
     2  6X,'AV WATER= ',F10.6)
      WRITE(25,108)
108   FORMAT(75('.'))
      TYPE *,TOUT,(U(J,1),J=1,7,6)
60    CONTINUE
      WRITE(25,19)
19    FORMAT(//,'THE GRID POINTS ARE(r IN m.):')
      IF(NPTS.EQ.12) GO TO 693
      WRITE(25,21) (X(I),I=1,NPTS)
21    FORMAT(10X,6F8.3)
      GO TO 692
693   WRITE(25,109) (X(I),I=1,NPTS)
109   FORMAT(10X,12F8.3)
C     ..........................................
C       OUTPUT FOR PLOTTING
C     ..........................................
692   WRITE(55,*) (X(I),I=1,NPTS),(TX(J),J=1,L)
```

```
      WRITE(55,*) (((RY(I,J,K),I=1,9),J=1,NPTS),K=1,L)
      WRITE(55,*) ((TY(I,J),I=10,14),J=1,L)
      STOP
99    TYPE *,IFAIL
100   TYPE *,IFAIL,TOUT
            STOP
      END
C     ************************************************
CC    SUBROUTINE PDEF
C     ************************************************
      SUBROUTINE  PDEF(NPDE,X,T,UX,DUX,F,G,C)
      DIMENSION C(NPDE),F(NPDE),DUX(NPDE),G(NPDE),UX(NPDE),
     1  AO(5),EO(5),FK(5),BK(5),EK(5),DS(5),DH(5),AC(5),EC(5)
      OPEN(UNIT=15,DEVICE='DSK',FILE='NYL.IN')
      READ(15,*) ICOND,IVEL,BB,TK,DD,B,IT,JJ,U10
      ITER=ITER+1
C     TYPE *,ITER,T,X,UX(1),DUX(1)
      DO 35 I=1,6
      C(I)=1.
      IF(IVEL.EQ.0)  GO TO 34
      C(I)=2.*(1.-(X/B)*(X/B))

      IF(X.EQ.B) C(I)=1.E-04
34          G(I)=0.5E-20
35    CONTINUE
      RHO=1000.0/(1.0065+0.0123*UX(1)+(UX(7)-495.)*(0.00035+
     2   0.00007*UX(1)))
      CP=0.6593*UX(1)/U10+(1.-UX(1)/U10)*(486.1+0.337*UX(7))
     3   /1000.
      C(7)=RHO*CP*2.*(1.-(X/B)*(X/B))
      IF(X.EQ.B) C(7)=1.E-04*RHO*CP
      IF(IVEL.EQ.0) C(7)=RHO*CP
      G(7)=TK
      AO(1)=5.9874E+05
      AO(2)=1.8942E+10
      AO(3)=2.8558E+09
      AO(4)=8.5778E+11
      AO(5)=2.5701E+08
      EO(1)=1.988E+04
      EO(2)=2.3271E+04
      EO(3)=2.2845E+04
      EO(4)=4.2E+04
      EO(5)=2.13E+04
      DH(1)=1.918E+03
      DH(2)=-5.9458E+03
      DH(3)=-4.0438E+03
      DH(4)=-9.6E+03
      DH(5)=-3.1691E+03
      DS(1)=-7.8846
      DS(2)=9.4374E-01
      DS(3)=-6.9487
      DS(4)=-14.52
      DS(5)=5.8265E-01
C     ................................................
      AC(1)=4.3075E+07
      AC(2)=1.2114E+10
      AC(3)=1.6377E+10
      AC(4)=2.3307E+12
      AC(5)=3.011E+09
      EC(1)=1.8806E+04
      EC(2)=2.067E+04
      EC(3)=2.0107E+04
      EC(4)=3.74E+04
      EC(5)=2.04E+04
      UR=1.987
      DO 11 I=1,5
      FK(I)=AO(I)*EXP(-EO(I)/(UR*UX(7)))+AC(I)*EXP(-EC(I)/(UR*
     1  UX(7)))*UX(3)
      EK(I)=EXP(DS(I)/UR-DH(I)/(UR*UX(7)))
      BK(I)=FK(I)/EK(I)
11    CONTINUE
```

```
      P2=UX(2)
      P3=UX(2)
      APR=(UX(3)-2.*UX(2))
      IF(APR.LT.0.0) APR=0.0
      IF(APR.LT.0.0) P2=0.0
      IF(APR.LT.0.0) P3=0.0
C     ..................................................................
      F(1)=-FK(1)*UX(1)*UX(6)+BK(1)*UX(2)-FK(3)*UX(1)*UX(3)
     4   +BK(3)*(UX(3)-UX(2))
C     ..................................................................
      F(2)=FK(1)*UX(1)*UX(6)-BK(1)*UX(2)-2.*FK(2)*UX(2)*UX(3)
     1   +2.*BK(2)*UX(6)*(UX(3)-UX(2))-FK(3)*UX(2)*UX(1)+
     2     BK(3)*P2-FK(5)*UX(2)*UX(5)+BK(5)*P3
C     ..................................................................
      F(3)=FK(1)*UX(1)*UX(6)-BK(1)*UX(2)-FK(2)*UX(3)*UX(3)+
     1     BK(2)*UX(6)*(UX(4)-UX(3))+FK(4)*UX(5)*UX(6)-
     2     BK(4)*P2
C     ..................................................................
      F(4)=FK(1)*UX(1)*UX(6)-BK(1)*UX(2)+FK(3)*UX(1)*UX(3)-
     1     BK(3)*(UX(3)-UX(2))+2.*FK(5)*UX(5)*UX(3)-2.*BK(5)*
     2     APR+2.*FK(4)*UX(6)*UX(5)-2.*BK(4)*P2
C     ..................................................................
      F(5)=-FK(4)*UX(5)*UX(6)+BK(4)*P2-FK(5)*UX(5)*UX(3)+
     1     BK(5)*APR
C     ..................................................................
      F(6)=-FK(1)*UX(1)*UX(6)+BK(1)*UX(2)+FK(2)*UX(3)*UX(3)-
     1     BK(2)*UX(6)*(UX(4)-UX(3))-FK(4)*UX(5)*UX(6)+BK(4)
     2     *P2
C     ..................................................................
      F1=-DH(1)*(FK(1)*UX(1)*UX(6)-BK(1)*UX(2))
     1   -DH(2)*(FK(2)*UX(3)*UX(3)-BK(2)*UX(6)*(UX(4)-UX(3)))
     2   -DH(3)*(FK(3)*UX(1)*UX(3)-BK(3)*UX(3)+BK(3)*UX(2))
     3   -DH(4)*(FK(4)*UX(5)*UX(6)-BK(4)*P2)
     4   -DH(5)*(FK(5)*UX(5)*UX(3)-BK(5)*APR)
C     ..................................................................
C     DO 12 I=1,7
C     IF(UX(1).LT.0.0) UX(1)=8.8
C12   CONTINUE
      IF(ICOND.EQ.0) GO TO 55
      F(7)=F1*RHO/1000.0
      GO TO 66
55    F(7)=0.
66    RETURN
      END
C     ******************************************************************
C     SUBROUTINE BNDY
C     ******************************************************************
      SUBROUTINE BNDY(NPDE,T,UX,IBND,P,Q,R)
      DIMENSION  P(NPDE),Q(NPDE),R(NPDE),UX(NPDE)
      OPEN(UNIT=15,DEVICE='DSK',FILE='NYL.IN')
      READ(15,*) ICOND,IVEL,UO,TK,TH
      IF(ICOND.EQ.1) UO=0.E+0
      IF(IBND.NE.0) GO TO  20
C     LEFT HAND B.C.
      DO 44 I=1,NPDE
      P(I)=0.
      Q(I)=1.
      R(I)=0.
44    CONTINUE
      RETURN
C     RIGHT HAND B.C.
20    CONTINUE
      DO 55 I=1,6
      P(I)=0.
      Q(I)=1.
      R(I)=0.
55    CONTINUE
      P(7)=0.
      Q(7)=-TK
```

```
      R(7)=UO*(UX(7)-TM)
C     R(7)=0.
      RETURN
      END
C     ***********************************************
```

```
C     ***********************************************
CCCC
      ORTHOGONAL COLLOCATION: METHOD IV
C     ***********************************************
      DIMENSION A(20,20),B(20,20),WT(10),COL(10),Y(70),W(70,90),RX(10)
      COMMON /OUT/H,IT,X,N,WT
      COMMON A,ICOL,COL,U10,P,RX,UO,TW,R,TK,U70,ICOND,IVEL
      EXTERNAL FCN,OUTPUT,PEDERV
      OPEN(UNIT=15,DEVICE='DSK',FILE='NOC.IN')
      OPEN(UNIT=39,DEVICE='DSK',FILE='NOC.OUT')
      OPEN(UNIT=6,DEVICE='DSK',FILE='ERR.OUT')
      OPEN(UNIT=45,DEVICE='DSK',FILE='COLL.OUT')
      READ(15,*) ICOND,IVEL,ICOL,
     1           UO,TK,TW,
     2              R,XEND,
     3           U10,U20,U30,U40,U50,U60,U70
      CALL COLLOC(ICOL,COL,A,B,WT)
C     TYPE *,(COL(I),I=1,ICOL+1)
      WRITE(45,*)((A(I,J),J=1,ICOL+1),I=1,ICOL+1)
      WRITE(45,*)((B(I,J),J=1,ICOL+1),I=1,ICOL+1)
C     TYPE *,(WT(I),I=1,ICOL+1)
      IF(ICOND.EQ.0) WRITE(39,17)
      IF(ICOND.EQ.1) WRITE(39,18)
      IF(ICOND.EQ.2) WRITE(39,19)
17    FORMAT(15X,50('*'),///,30X,'ISOTHERMAL REACTOR',///,15X,50('*').
     1   //)
18    FORMAT(15X,50('*'),///,30X,'ADIABATIC REACTOR',///,15X,50('*'),
     1   //)
19    FORMAT(15X,50('*'),///,22X,'TUBULAR REACTOR WITH HEAT
     1  TRANSFER',///,15X,50('*'),//)
      WRITE(39,20) R,ICOL,U10,U40,U20,U50,U70,U30,U60
20    FORMAT(5X,'RADIUS=',F6.2,5X,'NO OF COLL POINTS=',I3,/,
     1  5X,'MONOMER=',F8.4,5X,'1st MOMENT=',F8.4,5X,/,
     2  8X,'P(1)=',F8.4,10X,'DIMER=',F8.4,10X,'TEMP= ',F10.4,/,
     3  2X,'0th MOMENT=',F8.4,10X,'WATER=',F8.4,/,75('*'))
      TW=TW-273.
      IF(ICOND.EQ.2) WRITE(39,21) UO,TW
21    FORMAT(/,5X,'HEAT TRANSF COEFF= ',F10.5,2X,'KCAL/(SQ M)(HR)
     1  (DEG K)',/,10X,'COOLANT TEMP= ',F10.5,//,75('*'))
      TOL=1.E-06
      IRELAB=2
      MPED=0
      X=0.0
      H=0.5
      IT=0
      DO 33 J=1,ICOL+1
      RX(J)=2.*(1.-(COL(J))**2.)
      IF(RX(J).EQ.0.0) RX(J)=1.E-04
      IF(IVEL.EQ.0) RX(J)=1.0
33    CONTINUE
C     INITIAL CONDITION
      DO 22 I=1,(ICOL+1)
      Y(I)=U10
      Y(ICOL+1+I)=U20
      Y(ICOL*2+2+I)=U30

      Y(ICOL*3+3+I)=U40
      Y(ICOL*4+4+I)=U50
      Y(ICOL*5+5+I)=U60
      Y(ICOL*6+6+I)=U70/500.0
22    CONTINUE
      IF(IVEL.EQ.0) GO TO 23
      Y(ICOL+1)=0.670647
      Y(ICOL*2+2)=0.000234
      Y(ICOL*3+3)=0.043585
      Y(ICOL*4+4)=8.05944
      Y(ICOL*5+5)=0.034956
      Y(ICOL*6+6)=0.116415
23    IFAIL=0
      N=(ICOL+1)*7-1
```

```
      IW=18+N
      TW=TW/500.0
      CALL D02EBF(X,XEND,N,Y,TOL,IRELAB,FCN,MPED,
     1            PEDERV,OUTPUT,W,IW,IFAIL)
      WRITE(39,26) (COL(I),I=1,ICOL+1)
26    FORMAT(10X,10F8.3)
      IF(IFAIL.NE.0) GO TO 24
      STOP
24    TYPE *,IFAIL
      END
C     ***********************************************
CC    SUBROUTINE FCN
C     ***********************************************
      SUBROUTINE FCN(T,Y,F)
      DIMENSION A(20,20),Y(70),F(70),AO(5),EO(5),FK(5,10),
     1  BK(5,10),EK(5,10),DS(5),DH(5),AC(5),EC(5),COL(10),RHO(10)
     2  ,CP(10),RX(10),B(20,20),TP(10)
      COMMON A,ICOL,COL,U10,B,RX,UO,TW,R,TK,U70,ICOND,IVEL
      SUM=0.0
      DO 10 I=1,ICOL
10    SUM=SUM+A(ICOL+1,I)*Y(ICOL*6+6+I)
      TP(ICOL+1)=(-TK/R*SUM+UO*TW)/(UO+TK*A(ICOL+1,ICOL+1)/R)
      IF(ICOND.EQ.0) TP(ICOL+1)=U70/500.0
      DO 44 J=1,ICOL
44    TP(J)=Y(ICOL*6+6+J)
      DO 16 J=1,ICOL+1
      RHO(J)=1000.0/(1.0065+0.0123*Y(J)+(TP(J)*500.-495.)*
     3  (0.00035+0.00007*Y(J)))
      CP(J)=0.6593*Y(J)/U10+(1.-Y(J)/U10)*(486.1+0.337*TP(J)
     4  *500.0)/1000.0
16    CONTINUE
      AO(1)=5.9874E+05
      AO(2)=1.8942E+10
      AO(3)=2.8558E+09
      AO(4)=8.5778E+11
      AO(5)=2.5701E+08
      EO(1)=1.988E+04
      EO(2)=2.3271E+04
      EO(3)=2.2845E+04
      EO(4)=4.2E+04
      EO(5)=2.13E+04
      DH(1)=1.918E+03
      DH(2)=-5.9458E+03
      DH(3)=-4.0438E+03
      DH(4)=-9.6E+03
      DH(5)=-3.1691E+03
      DS(1)=-7.8846
      DS(2)=9.4374E-01
      DS(3)=-6.9487
      DS(4)=-14.52
      DS(5)=5.8265E-01
C     ...............................................................
      AC(1)=4.3075E+07
      AC(2)=1.2114E+10
      AC(3)=1.6377E+10
      AC(4)=2.3307E+12
      AC(5)=3.011E+09
      EC(1)=1.8806E+04
      EC(2)=2.067E+04
      EC(3)=2.0107E+04
      EC(4)=3.74E+04
      EC(5)=2.04E+04
      UR=1.987
      DO 11 I=1,5
      DO 11 J=1,ICOL+1
      FK(I,J)=AO(I)*EXP(-EO(I)/(UR*TP(J)*500.))+AC(I)*EXP
     1  (-EC(I)/(UR*TP(J)*500.0))*Y(ICOL*2+2+J)
      EK(I,J)=EXP(DS(I)/UR-DH(I)/(UR*TP(J)*500.0))
      BK(I,J)=FK(I,J)/EK(I,J)
11    CONTINUE
      DO 12 J=1,ICOL+1
```

```
      P2=Y(ICOL+1+J)
      P3=Y(ICOL+1+J)
      APR=(Y(ICOL*2+2+J)-2.*Y(ICOL+1+J))
      IF(APR.LT.0.0) APR=0.0
      IF(APR.LT.0.0) P2=0.0
      IF(APR.LT.0.0) P3=0.0
C     ..................................................................
      F(J)=-FK(1,J)*Y(J)*Y(ICOL*5+5+J)+BK(1,J)*Y(ICOL+1+J)
     1   -FK(3,J)*Y(J)*Y(ICOL*2+2+J)
     2   +BK(3,J)*(Y(ICOL*2+2+J)-Y(ICOL+1+J))
      F(J)=F(J)/RX(J)
C     ..................................................................
      F(ICOL+1+J)=FK(1,J)*Y(J)*Y(ICOL*5+5+J)-BK(1,J)*Y(ICOL+1+J)
     1   -2.*FK(2,J)*Y(ICOL+1+J)*Y(ICOL*2+2+J)
     2   +2.*BK(2,J)*Y(ICOL*5+5+J)*(Y(ICOL*2+2+J)-Y(ICOL+1+J))
     3   -FK(3,J)*Y(ICOL+1+J)*Y(J)+
     4     BK(3,J)*P2-FK(5,J)*Y(ICOL+1+J)*Y(ICOL*4+4+J)+BK(5,J)*P3
      F(ICOL+1+J)=F(ICOL+1+J)/RX(J)
C     ..................................................................
      F(ICOL*2+2+J)=FK(1,J)*Y(J)*Y(ICOL*5+5+J)-BK(1,J)*Y(ICOL+1+J)
     1   -FK(2,J)*Y(ICOL*2+2+J)*Y(ICOL*2+2+J)+
     2     BK(2,J)*Y(ICOL*5+5+J)*(Y(ICOL*3+3+J)-Y(ICOL*2+2+J))
     3   +FK(4,J)*Y(ICOL*4+4+J)*Y(ICOL*5+5+J)-
     4     BK(4,J)*P2
      F(ICOL*2+2+J)=F(ICOL*2+2+J)/RX(J)
C     ..................................................................
      F(ICOL*3+3+J)=FK(1,J)*Y(J)*Y(ICOL*5+5+J)-BK(1,J)*Y(ICOL+1+J)
     1   +FK(3,J)*Y(J)*Y(ICOL*2+2+J)-
     2     BK(3,J)*(Y(ICOL*2+2+J)-Y(ICOL+1+J))+2.*FK(5,J)*
     3   Y(ICOL*4+4+J)*Y(ICOL*2+2+J)-2.*BK(5,J)*
     4     APR+2.*FK(4,J)*Y(ICOL*5+5+J)*Y(ICOL*4+4+J)-2.*BK(4,J)*P2
      F(ICOL*3+3+J)=F(ICOL*3+3+J)/RX(J)
C     ..................................................................
      F(ICOL*4+4+J)=-FK(4,J)*Y(ICOL*4+4+J)*Y(ICOL*5+5+J)+BK(4,J)*P2
     1   -FK(5,J)*Y(ICOL*4+4+J)*Y(ICOL*2+2+J)+
     2     BK(5,J)*APR
      F(ICOL*4+4+J)=F(ICOL*4+4+J)/RX(J)
C     ..................................................................
      F(ICOL*5+5+J)=-FK(1,J)*Y(J)*Y(ICOL*5+5+J)+BK(1,J)*Y(ICOL+1+J)
     1   +FK(2,J)*Y(ICOL*2+2+J)*Y(ICOL*2+2+J)-
     2     BK(2,J)*Y(ICOL*5+5+J)*(Y(ICOL*3+3+J)-Y(ICOL*2+2+J))
     3   -FK(4,J)*Y(ICOL*4+4+J)*Y(ICOL*5+5+J)+BK(4,J)
     2     *P2
      F(ICOL*5+5+J)=F(ICOL*5+5+J)/RX(J)
C     ..................................................................
12    CONTINUE
      DO 13 JJ=1,ICOL
      P2=Y(ICOL+1+JJ)
      P3=Y(ICOL+1+JJ)
      APR=(Y(ICOL*2+2+JJ)-2.*Y(ICOL+1+JJ))
      IF(APR.LT.0.0) APR=0.0
      IF(APR.LT.0.0) P2=0.0
      IF(APR.LT.0.0) P3=0.0
      F1=-DH(1)*(FK(1,JJ)*Y(JJ)*Y(ICOL*5+5+JJ)-BK(1,JJ)*Y(ICOL+1+JJ))
     1   -DH(2)*(FK(2,JJ)*Y(ICOL*2+2+JJ)*Y(ICOL*2+2+JJ)-BK(2,JJ)*
     2   Y(ICOL*5+5+JJ)*(Y(ICOL*3+3+JJ)-Y(ICOL*2+2+JJ)))
     3   -DH(3)*(FK(3,JJ)*Y(JJ)*Y(ICOL*2+2+JJ)-BK(3,JJ)*
     4   Y(ICOL*2+2+JJ)+BK(3,JJ)*Y(ICOL+1+JJ))
     5   -DH(4)*(FK(4,JJ)*Y(ICOL*4+4+JJ)*Y(ICOL*5+5+JJ)-BK(4,JJ)*P2)
     6   -DH(5)*(FK(5,JJ)*Y(ICOL*4+4+JJ)*Y(ICOL*2+2+JJ)-BK(5,JJ)*APR)
C     ..................................................................
      SUM2=0.0
      DO 14 L=1,ICOL
14    SUM2=SUM2+B(JJ,L)*TP(L)
      F2=(TK/R**2.)*(SUM2+B(JJ,ICOL+1)*TP(ICOL+1))

      F(ICOL*6+6+JJ)=(F2+RHO(JJ)*F1/1000.0/500.0)/(RX(JJ)
     2   *RHO(JJ)*CP(JJ))
      IF(ICOND.EQ.0) F(ICOL*6+6+JJ)=0.0
13    CONTINUE
      RETURN
```

```
      END
C     ************************************************************
CCC   SUBROUTINE OUTPUT
C     ************************************************************
      SUBROUTINE   OUTPUT(XSOL,Y)
      DIMENSION A(20,20),B(20,20),Y(70),WT(10),RHO(10),CP(10),
     1  COL(10),RY(10,10,50),TY(15,50),U(10),UU(10),U1(10),U2(10),
     2  U3(10),U4(10),U5(10),U6(10),U7(10),RX(10)
      COMMON /OUT/H,IT,X,N,WT
      COMMON A,ICOL,COL,U10,B,RX,UO,TW,R,TK,U70,ICOND,IVEL
      TYPE *,IT
      IF(IT.EQ.0) GO TO 36
      DO 17 J=1,ICOL+1
      RY(1,J,IT)=Y(J)
      RY(2,J,IT)=Y(ICOL+1+J)
      RY(3,J,IT)=Y(ICOL*2+2+J)
      RY(4,J,IT)=Y(ICOL*3+3+J)
      RY(5,J,IT)=Y(ICOL*4+4+J)
      RY(6,J,IT)=Y(ICOL*5+5+J)
      RY(7,J,IT)=Y(ICOL*6+6+J)*500.0-273.
      RY(8,J,IT)=RY(4,J,IT)/RY(3,J,IT)
55    RY(9,J,IT)=(1.-RY(1,J,IT)/U10)*100.0
17    CONTINUE
      SUM1=0.0
      DO 25 K=1,ICOL
25    SUM1=SUM1+A(ICOL+1,K)*Y(ICOL*6+6+K)
      TLAST=(-TK/R*SUM1+UO*TW)/(UO+TK*A(ICOL+1,ICOL+1)/R)
      IF(ICOND.EQ.0) TLAST=U70/500.0
      RY(7,ICOL+1,IT)=TLAST*500.0-273.0

      DO 16 J=1,ICOL+1
      RHO(J)=1000.0/(1.0065+0.0123*Y(J)+(RY(7,J,IT)+273.0-495.)*
     3  (0.00035+0.00007*Y(J)))
      CP(J)=0.6593*Y(J)/U10+(1.-Y(J)/U10)*(486.1+0.337*(RY(7,J,IT)
     4  +273.0))/1000.0
16    CONTINUE
      SUM1=0.0;SUM2=0.;SUM3=0.;SUM4=0.;SUM5=0.;SUM6=0.;SUM7=0.
      SUM=0.;SUMM=0.
      DO 18 J=1,ICOL+1
      U(J)=RHO(J)*RX(J)
      UU(J)=U(J)*CP(J)
      U1(J)=U(J)*RY(1,J,IT)
      U2(J)=U(J)*RY(2,J,IT)
      U3(J)=U(J)*RY(3,J,IT)
      U4(J)=U(J)*RY(4,J,IT)
      U5(J)=U(J)*RY(5,J,IT)
      U6(J)=U(J)*RY(6,J,IT)
      U7(J)=U(J)*(RY(7,J,IT)+273.)*CP(J)
18    CONTINUE
      DO 19 J=1,ICOL+1
      SUM=SUM+WT(J)*U(J)
      SUMM=SUMM+WT(J)*UU(J)
      SUM1=SUM1+WT(J)*U1(J)
      SUM2=SUM2+WT(J)*U2(J)
      SUM3=SUM3+WT(J)*U3(J)
      SUM4=SUM4+WT(J)*U4(J)
      SUM5=SUM5+WT(J)*U5(J)
      SUM6=SUM6+WT(J)*U6(J)
      SUM7=SUM7+WT(J)*U7(J)
19    CONTINUE
      U1AV=SUM1/SUM
      TY(10,IT)=(1.-U1AV/U10)*100.
      TY(11,IT)=SUM4/SUM3
      TY(12,IT)=SUM6/SUM
      TY(13,IT)=SUM7/SUMM-273.
      TY(14,IT)=SUM5/SUM
      IF(XSOL.EQ.0.) GO TO 35
      WRITE(39,22) XSOL
22    FORMAT(11X,'TIME:',F5.2,2X,'Hr.',//)
      WRITE(39,23)  (RY(1,I,IT),I=1,ICOL+1)
      WRITE(39,39) (RY(2,I,IT),I=1,ICOL+1),(RY(3,I,IT),I=1,ICOL+1)
```

```
     1   ,(RY(4,I,IT),I=1,
     2   ICOL+1),(RY(5,I,IT),I=1,ICOL+1),(RY(6,I,IT),I=1,ICOL+1)
39   FORMAT(10X,'P(1) :',10F10.6,/,5X,'0th MOMENT:',10F10.6,/,
     1   5X,'1st MOMENT:',10F10.6,/,9X,'DIMER :',10F10.6,/,
     2   9X,'WATER :',10F10.6 )
23   FORMAT(8X,'MONOMER:',10F10.6)
     WRITE(39,24) (RY(7,I,IT),I=1,ICOL+1)
24   FORMAT(11X,'TEMP:',10F10.4)
     WRITE(39,26) (RY(8,I,IT),I=1,ICOL+1)
26   FORMAT(/,5X,'CHAIN LTH.:  ',10F10.3)
     WRITE(39,29) (RY(9,I,IT),I=1,ICOL+1)
29   FORMAT(4X,'CONVERSION :',10F10.3)
     WRITE(39,47) TY(10,IT),TY(13,IT),TY(11,IT),TY(14,IT),TY(12,IT)
47   FORMAT(/,7X,'AV CONV=  ',F10.6,5X,'AV TEMP= ',F10.4,/,
     1   2X,'AV CHAIN LTH=  ',F10.6,4X,'AV DIMER= ',F10.6,/,
     2   6X,'AV WATER=  ',F10.6)
     WRITE(39,27)
27   FORMAT(75('.'))
36   IT=IT+1
35   XSOL=X+FLOAT(IT)*H
     RETURN
     END
C    ************************************************
     SUBROUTINE PEDERV(T,Y,PW)
     DIMENSION Y(70),PW(70,70)
     RETURN
     END
     SUBROUTINE COLLOC(N,X,A,B,W)
C    IMPLICIT REAL *8 (A-H,O-Z)
     dimension qinv(10,10),z(10),c(10,10),d(10,10)
     dimension a(20,20),b(20,20),q(20,20),x(20),w(20)
     DIMENSION DIF1(10),DIF2(10),DIF3(10),ROOT(10)
     COMMON/ONE/ X1(20)
C    N=2
     NT=N+1
     BETA=0.0
     aa=2.
     CALL JCOBI(10,N,0,1,1.,BETA,DIF1,DIF2,DIF3,ROOT)
     ROOT(NT)=1.0
     DO 33 K=1,N
     X1(K)=(SQRT(ROOT(K)))
C    TYPE *,X1(K)
33   CONTINUE
     call coll(a,b,q,x,w,20,N,aa)
C    TYPE *,(X(I),I=1,NT)
CC   TYPE *,((A(I,J),J=1,NT),I=1,NT)
CC   TYPE *,((B(I,J),J=1,NT),I=1,NT)
C    TYPE *,(W(I),I=1,NT)
     RETURN
C    STOP
     END
     SUBROUTINE JCOBI (ND,N,NO,N1,AL,BE,DIF1,DIF2,DIF3,ROOT)
C    IMPLICIT REAL *8 (A-H,O-Z)

     DIMENSION DIF1(10),DIF2(10) ,DIF3(10),ROOT(10)

CCC   EVALUATION OF ROOTS AND  DERIVATIVES OF JAACOBI POLYNOMIALS
CCC   P(N)  (AL*BE); MACHINE ACCURACY 16 D;
CCC
CCC   FIRST EVALUATION OF COEFFICIENT IN RECURSION FORMULAS
CCC   RECURSION COEFFICIENTS ARE STORED IN DIF1 AND DIF2
C
     AB=AL+BE
     AD=BE-AL
     AP=BE*AL
     DIF1(1)=((AD/(AB+2.))+1.)/2.
     DIF2(1)=0.
     IF (N.LT. 2)  GO TO 15
     DO 10 I=2,N
     Z1=I-1
     Z=AB+2*Z1
```

```
      DIF1(I)=(((AB*AD)/(Z*(Z+2.)))+1.)/2.
      IF   (I .NE. 2) GO TO 11
       DIF2(I)=((AB+AP+Z1)/((Z+1)*Z*Z))
      GO TO 10
11    Z=Z*Z
      Y=Z1*(AB+Z1)
      Y=Y*(AP+Y)
      DIF2(I)=Y/((Z-1)*Z)
10    CONTINUE
C
C      ROOT DETERMINAATION BY NEWTON METHOD WITH SUPPRESSION OF PREVIOUS
C
15    X=0.
      DO 20  I=1,N
25    XD=0.
      XN=1.


      XD1=0.
      XN1=0.
      DO 30  J=1,N
      XP=(-DIF1(J)+X)*XN-DIF2(J)*XD
      XP1=(-DIF1(J)+X)*XN1-DIF2(J)*XD1+XN
      XD=XN
      XD1=XN1
      XN=XP
      XN1=XP1
30    CONTINUE
      ZC=1.
      Z=XN/XN1
      IF   (I .EQ. 1) GO TO 21
      DO 22 J=2,I

22    ZC=ZC-Z/(X-ROOT(J-1))
21    Z=Z/ZC
      X=X-Z
C     IF (DABS(Z).GT.1.D-09) GO TO 25
      IF(ABS(Z).GT.1.E-06) GO TO 25
      ROOT(I)=X
      X=X+.0001
20    CONTINUE
      DIF3(I)=0.
      RETURN
      END
      SUBROUTINE COLL(A,B,Q,X,W,ND,N,AA)
C     IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION A(ND,ND),B(ND,ND),Q(ND,ND),X(ND),W(ND)
      DIMENSION QINV(10,10),Z(10),C(10,10),D(10,10)
      COMMON/ONE/X1(20)
      N1=N+1
C
CC     THIS SUBROUTINE COMPUTES THE  MATRICES FOR ORTHOGONAL COLLOCATION
C     USING SYMMETRIC POLYNOMIALS
      DO 57 I=1,N
      X(I)=X1(I)
57    CONTINUE
      X(N1)=1.0
      DO 30 I=1,N1
      AI=I
      Z(I)=1./(2.*AI+AA-2.)
30    CONTINUE
      DO 35 I=1,N1
      Q(I,1)=1.
      QINV(I,1)=1.
      DO 35 J=2,N1

      Q(I,J)=X(I)**(2*J-2)
35    QINV(I,J)=Q(I,J)
      DO 40 J=1,N1
      CA=2.*J-2.
```

```
      DA=(2.*J-2.)*(2.*J+AA-4.)
      DO 40 I=1,N1
      C(I,J)=CA*X(I)**(2*J-3)
      D(I,J)=DA*X(I)**(2*J-4)
40    CONTINUE
      CALL INVR (QINV,N1,10)
      DO 50 I=1,N1
      W(I)=0.0
      DO 50 J=1,N1
      A(I,J) =0.0
      B(I,J)=0.0
      DO 45 K=1,N1
      A(I,J)=A(I,J)+C(I,K)*QINV(K,J)
      B(I,J)=B(I,J)+D(I,K)*QINV(K,J)
45    CONTINUE
      Q(I,J)=QINV(I,J)
      W(I)=W(I)+Z(J)*QINV(J,I)
50    CONTINUE
C     WRITE(60,12) (X(I),I=1,N1)
 12   FORMAT(3E16.5)
13    FORMAT('BIJ VALUES ARE ')

C      WRITE(60,12)
CC    WRITE(61,55)((A(I,J),I=1,N1),J=1,N1)
CC    WRITE(60,13)
CC    WRITE(62,55)((B(I,J),I=1,N1),J=1,N1)
55    FORMAT (9E16.5/)
      RETURN
      END

      SUBROUTINE INVR(A,N,NI)
C     IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION A(NI,NI),B(20),C(400)
      DO 5 J=1,N
      IND=N*(J-1)
      DO 5 I=1,N
      C(IND+I)=A(I,J)
5     CONTINUE
      CALL INVERT(N,1,C,B,1)
      DO 20 J=1,N
      DO 10 I=1,N
10    B(I)=0.
      B(J)=1.
      CALL INVSW(N,1,C,B,1)
      DO 15 I=1,N
      A(I,J)=B(I)
15    CONTINUE
20    CONTINUE
      RETURN
      END
      SUBROUTINE INVERT(N,NE,A,B,ITYPE)
C     IMPLICIT REAL *8 (A-H,O-Z)

      DIMENSION A(1000,3),B(N),A1(1000),B1(1000),C1(1000)
      GO TO (5,10,15),ITYPE
5     CALL DECOMP(N,A)
      RETURN
10    NP=(N-1)/NE+1
      PAUSE 14
      CALL LUDECO(NP,NE,A)
      RETURN
15    DO 20 K=1,N
      A1(K)=A(K,1)
      B1(K)=A(K,2)
      C1(K)=A(K,3)
20    CONTINUE
      CALL INVTRI(N,A1,B1,C1)
      DO 25 K=1,N
      A(K,1)=A1(K)
      A(K,2)=B1(K)
      A(K,3)=C1(K)
```

```
25      CONTINUE
        RETURN
        ENTRY INVSW
        GO TO (30,35,40) ,ITYPE
30      CALL SOLVE(N,A,B)
        RETURN
35      CALL FAS(NP,NE,N,A,B)
        RETURN
40      CALL SWEEP(N,A1,B1,C1,B)
        RETURN
        END
C
C
        SUBROUTINE SING(I)
C       IMPLICIT REAL *8 (A-H,O-Z)
        GO TO (5,15),I
5       WRITE(60,10)
10      FORMAT('MATRIX WITH ZERO ROW DECOMPOSE')
        RETURN
15      WRITE(60,20)
20      FORMAT('SINGULAR MATRIX IN DECOMPOSE. ZERO IN SOLVE ')
        RETURN
        END
        SUBROUTINE DECOMP(N,A)
C       IMPLICIT REAL *8 (A-H,O-Z)
        DIMENSION A(N,N)
        COMMON /DENSE/ IPS(201),SC(201)
C
C
        IF (N .EQ. 1) RETURN
        DO 25 I=1,N
        IPS(I)=I
        ROWNRM=0.0
        DO 10 J=1,N
        IF (ROWNRM-ABS(A(I,J))) 5,10,10
5       ROWNRM =ABS(A(I,J))
10      CONTINUE
        IF (ROWNRM) 15,20,15
15      SC(I)=1./ROWNRM
        GO TO 25
C
20      CALL SING(1)
        SC(I)=0.
25      CONTINUE
C
        NM1=N-1
        DO 65 K=1,NM1
        BIG=0.
        DO 35 I=K,N
        IP=IPS(I)
        SIZE=ABS(A(IP,K)*SC(IP))
        IF (SIZE-BIG) 35,35,30
30      BIG=SIZE
        IDXPIV=I
35      CONTINUE
        IF (BIG) 45,40,45
40      CALL SING(2)
        GO TO 65

45      IF (IDXPIV-K) 50,55,50
 50     J=IPS(K)
        IPS(K)=IPS(IDXPIV)
        IPS(IDXPIV)=J
55      KP=IPS(K)
        PIVOT=A(KP,K)
        KP1=K+1
        DO 60 I=KP1,N
        IP=IPS(I)
        EM=-A(IP,K)/PIVOT
        A(IP,K)=-EM
        DO 60 J=KP1,N
```

```
      A(IP,J)=A(IP,J)+EM*A(KP,J)
60    CONTINUE
65    CONTINUE
      KP=IPS(N)
      IF (A(KP,N)) 75,70,75
70    CALL SING(2)
75    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SOLVE(N,A,B)
C     IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION A(N,N),B(N)
      COMMON /DENSE/ IPS(201),SC(201)
C
C
      IF (N .GT. 1) GO TO 5
      B(1)=B(1)/A(1,1)
      RETURN
5     CONTINUE
      NP1=N+1
      IP=IPS(1)
      SC(1)=B(IP)
      DO 15 I=2,N
      IP=IPS(I)
      IM1=I-1
      SUM=0.
      DO 10 J=1,IM1
      SUM=SUM+A(IP,J)*SC(J)
10    CONTINUE
      SC(I)=B(IP)-SUM
15    CONTINUE
      IP=IPS(N)
      SC(N)=SC(N)/A(IP,N)
      DO 25 IBACK=2,N
      I=NP1-IBACK
      IP=IPS(I)
      IP1=I+1
      SUM=0.
      DO 20 J=IP1,N
      SUM=SUM+A(IP,J)*SC(J)
20    CONTINUE
      SC(I)=(SC(I)-SUM)/A(IP,I)
25    CONTINUE
      DO 30 I=1,N
30    B(I)=SC(I)
      RETURN
      END
C
C
      SUBROUTINE FAS(NP,NE,NT,A,B)
C     IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION A(NP,NP,NE),B(NT)
C
      NP1=NP
      DO 10 L=1,NE
      DO 10 I=2,NP1
      I2=(L-1)*(NP-1)+I
      S=0.
      I1=I-1
      DO 5 J=1,I1
      J2=I2-I+J
      S=S+A(I,J,L)*B(J2)
5     CONTINUE
      B(I2)=B(I2)-S
10    CONTINUE
C
      DO 25 L1=1,NE
      L=NE-L1+1
      IF (L.NE.NE) GO TO 15
```

```
      B(NT)=B(NT)/A(NP,NP,NE)
15    N1=NP-1
      DO 25 K=1,N1
      I=N1+1-K
      I2=(L-1)*(NP-1)+I
      M=I+1
      N2=N1+1
      S=0.
      DO 20 J=M,N2
       J2=(L-1)*(NP-1)+J
20     S=S+A(I,J,L)*B(J2)
       B(I2)=(B(I2)-S)/A(I,I,L)
25    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE LUDECO(NP,NE,A)
C     IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION A(NP,NP,NE)
C
      N1=NP-1
      DO 10 L=1,NE
        DO 5 K=1,N1
         K1=K+1
         DO 5 I=K1,NP
          S=A(I,K,L)/A(K,K,L)
          A(I,K,L)=S
          DO 5 J=K1,NP
            A(I,J,L)=A(I,J,L)-S*A(K,J,L)
5     CONTINUE
      IF  (L.EQ.NE) RETURN
      A(1,1,L+1)=A(NP,NP,L)
10    CONTINUE
      END
C
C
      SUBROUTINE INVTRI(N,A,B,C)
C     IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION A(N),B(N),C(N)
C
C
      DO 5 L=2,N
        S=A(L)/B(L-1)
        B(L)=B(L)-S*C(L-1)
5     A(L)=S
      RETURN
      END
C
C
      SUBROUTINE SWEEP(N,A,B,C,D)
C     IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION A(N),B(N),C(N),D(N)
C
      DO 5 L=2,N
5     D(L)=D(L)-A(L)*D(L-1)
C
      D(N)=D(N)/B(N)
      DO 10 L=2,N
       K=N-L+1
10    D(K)=(D(K)-C(K)*D(K+1))/B(K)
      RETURN
      END
C
C
```